National Institute of Technology Srinagar

COURSES & SYLLABI
(Batch 2019 onwards)

Computer Science & Engineering Department

Department of Computer Science & Engineering
National Institute of Technology Srinagar.

Introduction

The Department of Computer Science & Engineering started functioning from 2007. It offers four year B.Tech degree course in Computer Science & Engineering. The initial intake in 2007 was 40 and has now increased to 60. The degree is awarded after completion of a minimum of 200 credits. Common courses for 50 credits are offered to students of all branches in a common first year spread over 2 semesters. Courses for the remaining 150 credits are offered to students during a span of three years spread over 6 semesters.

The first Board of Studies (BoS) meeting of the B.Tech Computer Science & Engineering Course was held in May 2008. In the meeting courses to be taught at 3rd and 4th Semester level were only approved. The second Board of Studies (BoS) was held on November 2009. The third BOS was held on 16-06-2014 and fourth BOS on 06-03-2015. In this meeting, course scheme of B.Tech Computer Science & Engineering degree course from 3rd to 8th Semesters was prepared, examined, revised, formulated and approved for batch starting from 2014. The scheme of courses has been designed such that at least 50% of the courses are offered by Department of Computer Science & Engineering. The remaining 50% courses are interdisciplinary and are offered by Departments of Information Technology, Department of Electronics & Communication Engineering, Electrical Engineering, Mathematics and Humanities Departments.

Other main features of the scheme are:-

1. Courses offered are 2 credits, 3 credits or 4 credits.
2. One hour lecture/tutorial has been assigned 1 credit weightage.
3. Two hour laboratory per week has also been assigned 1 credit weightage.
4. A continuous evaluation scheme is used to evaluate the students for each course. The evaluation is as under:

Midterm	30 marks
Class Assessment	10 marks
Major Exam	60 marks.

5. Grades are allotted to the students as per the following scheme:

Marks	Grades	Points
0 to 39	F	4
40 to 50	C	5
51 to 60	C^+	6
61 to 70	B	7
71 to 80	B^+	8
81 to 90	A	9
91 to 100	A^+	10

6. At the end of each semester a cumulative grade point average (CGPA) is calculated for the courses taken by a student.

National Institute of Technology Srinagar

Hazratbal, Srinagar, Kashmir, 190006 India.

Computer Science & Engineering Department

Table of Contents

Department of Computer Science & Engineering
National Institute of Technology Srinagar

Course Structure for B.Tech. Computer Science & Engineering

(Batch 2019 onwards)

Semester: 3rd

S.No.	Subject	Code	L	T	P	Credits
1	Object Oriented Programming	CST201	3	1	0	4
2	Object Oriented Programming -Lab	CSL202	0	0	2	1
3	Internet & Web Technologies		0	0	4	2
4	Signals & Systems		3	1	0	4
5	Basic Electronics		3	1	0	4
6	Basic Electronics Lab		0	0	2	1
7	Database Management Systems	CST203	3	1	0	4
8	Database Management Systems - Lab	CSL204	0	0	2	1
9	Discrete Mathematics		3	1	0	4
	Total Credits					**25**

Semester: 3rd - Other Department Courses - Information Technology

S.No.	Subject	Code	L	T	P	Credits
1	Object Oriented Programming	CST201	3	1	0	4
2	Object Oriented Programming -Lab	CSL202	0	0	2	1

Semester: 4th

S.No.	Subject	Code	L	T	P	Credits
1	Data Structures	CST250	3	1	0	4
2	Data Structures – Lab	CSL251	0	0	2	1
3	Introduction to Probability Theory & Statistics		3	0	0	3
4	Digital Electronics & Logic Design		3	1	0	4
5	Digital Electronics & Logic Design – Lab		0	0	2	1
6	Software Engineering		3	1	0	4
7	Communication Systems		3	1	0	4
8	Communication Systems – Lab		0	0	2	1
9	Control Systems		3	0	0	3
	Total Credits					**25**

Semester: 4th - Other Department Courses - Information Technology

S.No.	Subject	Code	L	T	P	Credits
1	Data Structures	CST250	3	1	0	4
2	Data Structures –Lab	CSL251	0	0	2	1

Semester: 5th

S.No.	Subject	Code	L	T	P	Credits
1	Computer Organization & Architecture	CST305	3	1	0	4
2	Design & Analysis of Algorithms	CST306	3	1	0	4
3	Microprocessor	CST307	3	0	0	3
4	Microprocessor - Lab	CSL308	0	0	2	1
5	Operating Systems	CST309	3	1	0	4
6	Python Programming	CST310	3	1	0	4
7	Python Programming - Lab	CSL311	0	0	2	1
8	Data Communication	CST312	3	1	0	4
	Total Credits					**25**

Semester: 5th - Other Department Courses

- Electronics & Communication Engineering

S.No.	Subject	Code	L	T	P	Credits
1	Data Structures	CST313	3	1	0	4
2	Data Structures Lab	CSL314	0	0	2	1

- Information Technology

S.No.	Subject	Code	L	T	P	Credits
1	Design & Analysis of Algorithms	CST306	3	1	0	4

Semester: 6th

S.No.	Subject	Code	L	T	P	Credits
1	Artificial Intelligence	CST352	3	1	0	4
2	Artificial Intelligence - Lab	CSL353	0	0	2	1
3	Computer Networks	CST354	3	1	0	4
4	Computer Networks - Lab	CSL355	0	0	2	1
5	Theory of Computation	CST356	3	1	0	4
6	Computer Graphics	CST357	3	0	0	3
7	Computer Graphics -Lab	CSL358	0	0	2	1
8	Java Programming	CST359	2	0	2	3
9	Practical Training & Tour	CSI360		-		1
10	Elective I / Swayam Online Course	CST0XX	3	0	0	3
	Total Credits					**25**

Semester: 7th

S.No.	Subject	Code	L	T	P	Credits
1	Compiler Design	CST415	3	1	0	4
2	Compiler Design - Lab	CSL416	0	0	2	1
3	Network Security	CST417	3	1	0	4

4	Network Security - Lab	CSL418	0	0	2	1
5	Pre-Project	CSP419	0	0	4	2
6	Seminar	CSS420	0	0	2	1
7	Elective II	CST0XX	3	0	0	3
8	Elective III	CST0XX	3	0	0	3
9	Elective IV/ Swayam Online Course	CSE0XX	3	0	0	3
10	Operations Research & Optimization		3	0	0	3
	Total Credits					**25**

Semester: 7th - Other Department Courses

- Electronics & Communication Engineering (M.Tech.)

S.No.	Subject	Code	L	T	P	Credits
1	Internet & Web Design	CSL501	1	0	2	2
2	RDBMS	CST502	2	0	2	3

- Information Technology (Elective)

S.No.	Subject	Code	L	T	P	Credits
1	Compiler Design	CST415	3	0	0	3
2	Compiler Design - Lab	CSL416	0	0	2	1

Semester: 8th

S.No.	Subject	Code	L	T	P	Credits
1	Project	CSP460	0	0	10	10
2	Elective V	CST0XX	3	0	0	3
3	Elective VI	CST0XX	3	0	0	3
4	Elective VII	CST0XX	3	0	0	3
5	Elective VIII	CST0XX	3	0	0	3
6	Industrial Organization & Management		3	0	0	3
	Total Credits					**25**

List of Electives

S.No.	Subject	Code
1.	Simulation & Modeling	CST001
2.	Graph Theory	CST002
3.	Digital Signal Processing	CST003
4.	Multimedia Technology	CST004
5.	Logic Programming	CST005
6.	Embedded Systems	CST006
7.	Advanced Java & Android Programming	CST007
8.	System on Chip (SoC)	CST008
9.	Advanced Internet Technologies	CST009
10.	Wireless Communication	CST010
11.	Fault Tolerant Computing	CST011
12.	Image Processing	CST012
13.	System Design using HDL	CST013
14.	Real Time Systems	CST014
15.	Unix & Shell Programming	CST015
16.	High Speed Networks	CST016
17.	Advanced Algorithms	CST017
18.	Reconfigurable Computing	CST018
19.	Computer Vision	CST019
20.	Advanced Computer Networks	CST020
21.	Advanced Computer Graphics	CST021
22.	Advanced DBMS	CST022
23.	Advanced Computer Architecture	CST023
24.	Advanced Compilation Techniques	CST024
25.	Principles of Cryptography	CST025
26.	Neural Networks	CST026
27.	Pervasive Computing	CST027
28.	Distributed and Parallel Computing	CST028
29.	Cloud Computing	CST029
30.	Software Project Management	CST030
31.	Big Data	CST031
32.	Cyber laws and Forensics	CST032
33.	Expert Systems	CST033
34.	Mobile Computing	CST034
35.	Green Computing	CST035
36.	Introduction to Robotics	CST036
37.	Data Analytics	CST037
38.	Computational Biology	CST038
39.	Special topics in Computer Science	CST039
40.	System & Network Administration	CST040
41.	Pattern Recognition	CST041

42.	Natural Language Processing	CST042
43.	Quantum Computing	CST043
44.	Deep Learning	CST044
45	Introduction to Data Science	CST045
46.	Internet of Things (IoT)	CST046
47.	Advanced Cryptography	CST047
48.	Data Mining	CST048
49.	Advanced Graph Algorithms	CST049
50.	Advanced Java	CST050
51.	Numerical Methods	Maths

List of Swayam Online Courses

S.No	Course Name	Course Duration
1.	Animations	15 weeks
2.	Web based Technologies & multimedia	15 weeks
3.	Applied Natural Language Processing	12 weeks
4.	Art of C Programming	12 weeks
5.	Artificial Intelligence	12 weeks
6.	Artificial Intelligence Search Methods for Problem Solving	12 weeks
7.	Bandit Algorithm	12 weeks
8.	Computer Fundamentals	12 weeks
9.	Computational Number Theory and Algebra	12 weeks
10.	Computer Networks	12 weeks
11.	Computer Architecture &Organization	12 weeks
12.	Cyber Security	12 weeks
13.	Cyber Security	12 weeks
14.	Concrete Technology	12 weeks
15.	Data Mining	12 weeks
16.	Data Structure and Algorithm in Java	12 weeks
17.	Deep Learning	12 weeks
18.	Deep Learning in Computer Vision	12 weeks
19.	Design and Analysis of Algorithms	12 weeks
20.	Discrete Math	12 weeks
21.	Ecommerce Technologies	12 weeks
22.	Electricity and Safety Measures	12 weeks
23.	Energy Resources and Conversion Processes	14 weeks
24.	Fundamentals of computer systems	12 weeks
25.	Introduction to IT	12 weeks
26.	Linux for Sys-Ads	12 weeks
27.	Linux Bash	12 weeks
28.	Linux Operating System	12 weeks

National Institute of Technology Srinagar

3rd Semester

Computer Science & Engineering

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Object Oriented Programming	**Semester**	3rd	
Department	Computer Science & Engineering	**Course Code**	CST201	
Credits	04	**L**	**T**	**P**
Course Type	Theory	3	1	0

Course Objectives

- Understand the basic principles and application of object oriented programming with main focus on C++ by getting familiar with the keywords, tokens and its other control structures.
- Demonstrate the function, class, object concept and implement & relate them with day today live examples in C++ Code.
- Understand how to create and delete instances of classes and Implement the same along with its different types using C++.
- Demonstrate and Implement the main features of Object Oriented Programming Languages along with the pointer concept in relation with real world examples using code in C++.
- Understand and implement generic classes, functions, files and exception handling using suitable examples in C++ for developing software products.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction:** Object oriented thinking: Need for OOP Paradigm, Procedural programming vs Object Oriented Programming, object oriented concepts. Benefits, Languages and Applications of OOPs. **Tokens, Expressions and Control Structures:** Tokens, Keywords, Identifiers & Constants, Basic Data types, User-defined Data types, Derived Data Types, Memory Management Operators, Manipulators, Expressions, Operator Overloading, Control Structures**.**	3
2.	**Functions in C++:** Main function, function prototyping, call by reference, inline functions, default functions, function overloading. **Classes and Objects:** Specifying a class, defining member functions, private memberfunctions,array within a class, memory allocation for objects, arrays of objects,Access Specifiers, scope resolution operator, objects as function arguments, returning objects, pointers to members, local classes, Friend Functions.	3
3.	**Constructors & Destructors:** Constructor function, types of constructors:default,parameterized and copyconstructor,constructor overloading, constructor with default parameter, dynamic initialization ofobjects,destructor.	3

	Strings: Creating and manipulating string objects, accessing characters in strings, comparing and swapping.	
4.	**Operator Overloading & Type Conversion:** Definition & Rules of overloading Operators, Overloading Binary & Unary Operators.Data Conversion: Basic to User Defined, User defined to basic, Conversion from one user-defined to other. **Inheritance:** Definition, single, multilevel, multiple, hierarchical and hybrid inheritance, virtual base classes, abstract classes. **Pointers, Virtual Functions and Polymorphism:** Pointers, Pointers to Objects and derived classes, early vs. late bindingvirtual functions, pure virtual functions.	3
5.	**Templates:** Class templates, function templates, overloading of function templates, member function templates. Overview of Standard Template Library, Containers, Algorithms, Iterators, Other STL Elements, Container Classes, General Theory of Operation, Vectors. **Exception Handling:** Exceptions, handling various types of Exceptions(*Try,Catch, throw)* including any universal exception. **File and Stream Handling:**Classes for file stream operations, opening and closing files, File opening modes, file Pointers, Error handling during file operations, File I/O with Member Functions, Overloading the Extraction and Insertion Operators, memory as a Stream Objectcommand line arguments.	3
Text Books		
1.	Robert Lafore, "Object Oriented Programming in Turbo C++", Galgotia Publications,	
2.	Balagurusamy, "Object Oriented programming with C++", Tata McGraw Hill.	
References		
1.	BjarneStrustrup, "The C++ programming Language", Addison Wesley,	
2.	Booch, "Object Oriented Analysis and Design with Applications, Addison Wesley.	
3.	Chair H. Pappas & William H. Murray, "The Complete Reference Visual C++", TMH.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Object Oriented Programming lab	**Semester**	3rd	
Department	Computer Science & Engineering	**Course Code**	CST202	
Credits	01	**L**	**T**	**P**
Course Type	Lab	0	0	2

Course Objectives

- The student should be able to explain the fundamental properties of the C++ language.
- The student should be able to combine the elements of the C++ language in developing structured programs.
- The student should be able to demonstrate the skills necessary to correctly compile, debug, and test programs in C++.

Learning Outcomes

At the end of the course student will be able to:

- Apply C++ features to program design and implementation
- Explain object-oriented concepts and describe how they are supported by C++
- Use C++ to demonstrate practical experience in developing object-oriented solutions
- Analyse a problem description and design and build object-oriented software using good coding practices and techniques
- Use common software patterns in object-oriented design and recognise their applicability to other software development contexts.

Course Synopsis

The objective of the lab course to familiarise students with C++ concepts.

Course Outline / Content

Unit	Topics	Week
Lab #1, #2	Simple programs in C++ using inbuilt input/output functions.	2
Lab #2	Function overloading, default arguments in C++.	1
Lab #3	Simple class design in C++, namespaces, objects creations.	1
Lab #4	Class design in C++ using dynamic memory allocation.	1
Lab #5	Constructor, Destructor, copy constructors.	2
Lab #6	Operator overloading, friend functions.	1
Lab #7	Overloading assignment operator, type conversions.	2
Lab#8	Inheritance, run-time polymorphism.	1
Lab #9	Template design in C++.	1
Lab #10	Interfaces and Inheritance.	1
Lab #11	File and Exception handling.	1

Text Books

1. Robert Lafore, "Object Oriented Programming in Turbo C++", Galgotia Publications,
2. Balagurusamy, „Object Oriented programming with C++", Tata McGraw Hill.

References	
1.	BjarneStrustrup, “The C++ programming Language”, Addison Wesley,
2.	Booch, “Object Oriented Analysis and Design with Applications, Addison Wesley.
3.	Chair H. Pappas & William H. Murray, “The Complete Reference Visual C++”, TMH.

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Database Management Systems	**Semester**	3rd	
Department	CSE	**Course Code**	CST203	
Credits	04	**L**	**T**	**P**
Course Type	Theory	3	1	0

Course Objectives

- Learn and practice data modelling using the entity-relationship and developing database designs.
- Apply normalization techniques to normalize the database
- Understand the use of Structured Query Language (SQL) and learn SQL syntax.
- Understand the needs of database processing and learn techniques for controlling the consequences of concurrent data access.
- Learn about the basics data recovery techniques.

Learning Outcomes

Upon successful completion of the course, the student will be able to:

- Differentiate database systems from file systems by enumerating the features provided by database systems and describe each in both function and benefit.
- Define the terminology, features, classifications, and characteristics embodied in database systems.
- Demonstrate an understanding of the relational data model.
- Transform an information model into a relational database schema and to use a data definition language and/or utilities to implement the schema using a DBMS.
- Formulate, using relational algebra, solutions to a broad range of query problems.
- Formulate, using SQL, solutions to a broad range of query and data update problems.
- Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.
- Use an SQL interface of a multi-user relational DBMS package to create, secure, populate, maintain, and query a database.
- Understand the different query processing and transaction management techniques.
- Recover a database from a possible failure.

Course Synopsis

Introduction to DBMS- Historical perspective, File Versus a DBMS, Advantages of DBMS. ER model, Relational Algebra, Relational Calculus and SQL- Queries. ACID properties, transactions, schedules and concurrent executionof transactions. Overviewof Query Evaluation, operator evaluation, Database Recovery, Failure classification, Recovery and atomicity.

Course Outline / Content

Unit	Topics	Week
1.	**Basic Concepts and Conceptual Database Design:** Database Users, Characteristics of the Database, Database Systems, Data Models, DBMS Architecture & Data Independence, Database Languages & Interfaces. Overview of Hierarchical, Network & Relational Data Base Management Systems. Data Modelling Using the Entity-Relationship Model – Entities, Attributes and Relationships, Cardinality of Relationships, Strong and Weak	3

	Entity Sets, Generalization, Specialization, and Aggregation, Translating your ER Model into Relational Model.Integrity constraints overrelations	
2.	**Relational Data Base Design and Oracle Architecture:** Functional Dependencies & Normalization for Relational Databases, Functional Dependencies, Normal Forms, Lossless Join & Dependency, Preserving Decomposition, Database Storage, Oracle Software Structures, Shared Database Access Mechanism, Database Protection. Case Study – ORACLE.	3
3.	**Relational Model, Languages & Systems:** Relational Data Model, Relational Model Concepts, Relational Model Constraints, Relational Algebra, SQL – A Relational Database Language, Data Definition & Manipulation in SQL, View and Queries in SQL, Specifying Constraints and Indexes in SQL, Practicing SQL commands using ORACLE. Case Study – ORACLE. **Query Processing:** Overviewof Query Evaluation, operator evaluation, Algorithms for relational operations- Selection operation, General selection condition, Projection operation, Join operation, set operation andaggregate operation, Evaluation of relational operations.	3
4.	**Transaction Management:** ACID properties, transactions, schedules and concurrent executionof transactions, Concurrency control- lock based protocol, Serializability, recoverability,dealing with deadlocks and Concurrency control without locking.	3
5.	**Database Recovery:** Failure classification, Recovery and atomicity, Log-based recoveryshadow paging and Advanced Recovery Techniques. Security and Authorization-Accesscontrol, Direct access control and Mandatory access control, Role of DBA.	2
Text Books		
1.	Database system Concept by Silberschatz and Korth 6th Edition	
2.	Elamsri, Navathe, Somayajulu and Gupta, “Fundamentals of Database Systems”, Pearson Education	
3.	Database Systems, Thomas Connolly, Carolyn Begg, Pearson 4th Edition	
References		
1.	Raghu Ramakrishnan, Johannes Gehrke, “Database Management Systems”, Tata McGraw Hill.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Database Management Systems Lab	**Semester**	3rd	
Department	Computer Science & Engineering	**Course Code**	CST204	
Credits	01	**L**	**T**	**P**
Course Type	Lab	0	0	2

Course Objectives

To implement the different concepts learned in the theory class of DBMS using embedded SQL and Oracle GUI.

Learning Outcomes

- Design and Implement a database schema
- Devise queries using DDL, DML, DCL and TCL commands.
- Develop application programs using PL/SQL
- Design and implement a project using embedded SQL and GUI.
- Apply modified components for performance tuning in open source software.

Course Synopsis

Familiarization of Oracle RDBMS, SQL*Plus, SQL- query structure, Exception Handling Compilation and Run-time, user-defined, Stored procedures.

Course Outline / Content

Unit	Topics	Week
Lab #1	1) Introduction to SQL, RDBMS. • Visualizing the architecture of RDBMS. • Different data types and its implementation.	1
Lab #2	1) SQL commands: • Implementation of Creating and managing SQL tables. • DDL(Data definition language): Implementation of Create, Alter, drop, rename, truncate, comment.	1
Lab #3	1) Basic Parts of speech in SQL • Implementation of Relational operators. • Implementation of Logical operators (ALL, AND, ANY, BETWEEN, EXISTS, IN, LIKE, NOT, OR, SUM) • SQL functions: (SUM, MAX, AVERAGE, LIKE)	1
Lab #4	1) Changing of Data in tables • DML(Data manipulation Language): Understanding the implementation of Select, Insert, Update, Delete, merge. 2) Retrival of data from the table • Understanding implementation of simple queries on single table only.	1
Lab #5	1) Implementation of constraints: Not null, Primary Key, Unique, Check, Foreign key) 2) Combining Tables and execution of queries on such tables: • Perform Join, inner join, outerjoin, natural join and subtypes of each. • Implementation of Advanced queries, subquery and grouping (Group by and having clause)	1

Lab #6	1) Understanding the dependence in queries, correlated queries using Existential quantifiers 2) Understanding difference in replacing IN with OUTER JOIN, EXISTS and NOT EXISTS.	1
Lab #7	1) Implementation of Security by assigning Privileges to database users DCL: (Data control Language) • Understanding the implementation of Grant, Revoke and views. TCL: (Transaction control Language): • Understanding the implementation of Begin, Commit, Rollback and Save point in transaction	1
Lab#8	1) Lab Project: Students are required to submit a case study	1
Text Books		
1.	James, Paul and Weinberg, Andy Oppel, "SQL: The Complete Reference", Tata McGraw Hill.	
2.	Michael McLaughlin, "Oracle Database 11g PL/SQL Programming", Oracle press.	

National Institute of Technology Srinagar

4th Semester

Computer Science & Engineering

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Data Structures	**Semester**	4^{th}	
Department	Computer Science & Engineering	**Course Code**	CST250	
Credits	04	**L**	**T**	**P**
Course Type	Theory	3	1	0

Course Objectives

- Understand the concept of ADTs(Abstract Data Types)
- Identify data structures suitable to solve problems
- Develop and analyze algorithms for stacks, queues along with their applications
- Develop algorithms for binary trees and graphs along with their applications
- Implement sorting and searching algorithms
- Implement symbol table using hashing techniques

Learning Outcomes

By the end of the course, the students will be able to :

- Design and analyze programming problem statements.
- Choose appropriate data structures and algorithms and use it to design algorithms for a specific problem.
- Understand the necessary mathematical abstraction to solve problems.

Course Synopsis

The course seeks to empower students with advanced programming concepts to enable them to become efficient programmers.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction:** Basic concept of data, structures and pointers. **Strings:** Representation, String operations, Implementing String.h library functions.	1
2.	**Arrays:** Representation, implementation, polynomial representation. Limitations.	1
3.	**Linear Data Structures: Linked Lists** Linked List and its comparison with array implementation. Types of Linked lists, Applications of Linked lists. Implementing Linked Lists using structures. Insertion, Deletion, Search, Print.	4
4.	**Stacks: Static and Dynamic** Implementation. Applications of Stacks. Prefix Postfix and Infix Expressions. Infix to postfix conversion, Expression evaluation, and expression trees.	3
5.	**Queues: Static and Dynamic** Implementation. Applications of Queues, Types of Queues, Array Implementation of Circular Queues,	3

	Search and Update Operations on Varieties of Linked Lists, Linked List Implementation of Stacks and Queues	
6.	**Recursion**: Recursion, Recursion and Stacks. Expression evaluation using stacks.	
7.	**Non-Linear Data Structures:** Introduction to Trees, Implementation of Trees, Binary Trees, Tree Traversals with an Application, Binary Search Trees (BSTs), Query and Update Operations on BSTs, static and dynamic implementation. Tree operations, insert, delete, and search. **Heaps:** Definition and Implementation of Max and Min Heap. Priority Queue ADT, Binary Heap Implementation and Applications of Priority Queues	4
8.	**Hashing:** Implementation of Dictionaries, Hash Function, Collisions in Hashing, Separate Chaining, Open Addressing.	2
9.	**Sorting Algorithms:** Stability and In Place Properties, Insertion Sort, Merge Sort, Quick Sort, Heap Sort, Lower Bound for Comparison Based Sorting Algorithms, **Linear Sorting Algorithms:** Counting Sort, Radix Sort, Bucket Sort	3
10.	**Graph Algorithms:** Graphs and their Representations, Graph Traversal Techniques: Breadth First Search (BFS) and Depth First Search (DFS), Applications of BFS and DFS, Minimum Spanning Trees (MST), Prim's and Kruskal's algorithms for MST, Connected Components, Dijkstra's Algorithm for Single Source Shortest Paths. Tree Traversals.	3
11.	**Storage Management:** Memory Management techniques, garbage collection.	1
Text Books		
1.	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Second Edition, PHI, 2009.	
2.	Ellis Horowitz, SartajSahni and SanguthevarRajasekaran, Fundamentals of Computer Algorithms, Second Edition, Universities Press, 2011.	
3.	Data Structures by Rajni Jindal	
4.	Data Structures - Schaum's Series	
References		
1.	Data Structures by Knuth	
2.	Data Structures by Farouzan	
3.	Data Structures using C and C++ by Langsam, Augestern, Tanenbaum.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Data Structures Lab	**Semester**		4th
Department	Computer Science & Engineering	**Course Code**		CST251
Credits	01	**L**	**T**	**P**
Course Type	Lab	0	0	2

Course Objectives

- Develop ADT for stack and queue applications
- Implement tree and graph algorithms
- Implement and analyse internal and external sorting algorithms
- Design and implement symbol table using hashing technique

Learning Outcomes

Basic concepts of data, linear lists, strings, arrays and orthogonal lists, representation of trees & graphs, storage systems, Arrays, Recursion, Stacks, Queues, Linked lists, Binary trees, General Trees, Tree Traversal, Symbol Table and Searching Techniques, Sorting Techniques, Graphs.

Course Synopsis

To enable a student to have a practical command over the concepts learned in the course.

Course Outline / Content

Unit	Topics	Week
1.	Implement singly and doubly linked lists.	1
2.	Represent a polynomial as a linked list and write functions for polynomial addition.	1
3.	Implement stack and use it to convert infix to postfix expression	1
4.	Implement array-based circular queue.	1
5.	Implement an expression tree. Produce its pre-order, in-order, and post-order traversals.	1
6.	Implement binary search tree.	1
7.	Implement priority queue using heaps	1
8.	Implement hashing techniques	2
9.	Implement various sorting techniques	2

Text Books

1.	Data Structures by Rajni Jindal
2.	Data Structures - Schaum‘s Series

References

1.	Data Structures by Knuth
2.	Data Structures by Farouzan
3.	Data Structures using C and C++ by Langsam, Augestern, Tanenbaum.

National Institute of Technology Srinagar

5th Semester

Computer Science & Engineering

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Computer Organization & Architecture	**Semester**	5^{th}	
Department	Computer Science & Engineering	**Course Code**	CST305	
Credits	04	**L**	**T**	**P**
Course Type	Theory	3	1	0

Course Objectives

- To understand the basics of computer organization: structure and operation of computers and their peripherals.
- To describe arithmetic and logical operations with integer and floating-point operands and their representation in computers and implement the Hardware for Arithmetic Operations.
- To study basic processing unit and organization of simple processor, concept of pipelining and other large computing systems.
- To study hierarchical memory systems including cache memories and virtual memory.
- To study different ways of communicating with I/O devices and standard I/O interfaces.

Learning Outcomes

Upon completion of the course the student will be able to:

- Understand the basic structure and operation of digital computer;
- Study the design of arithmetic and logic unit and implementation of fixed point and floating-point arithmetic operations;
- Study the two types of control unit techniques and the concept of pipelining;
- Study the hierarchical memory system including cache memories and virtual memory;
- Study the different ways of communicating with I/O devices and standard I/O interfaces.

Course Synopsis

Overview of basic digital building blocks; Number system; building blocks for the ALU; CPU buses; Concept of sub-routine; Memory organization; interrupts; VHDL concepts.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction:** Overview of basic digital building blocks; basic structure of a digital computer. Basic Performance Equation, Clock Rate, Performance Measurement. Number system and representation of information, arithmetic and logical operation, hardware implementation, Real numbers - fixed and floating point, IEEE754 representation.	3
2.	**CPU Subblock:** Datapath - ALU, Registers, CPU buses; Control path – microprogramming (only the idea), hardwired logic; External interface. Various addressing modes. Concept of sub-routine and sub-routine call. Use of stack for handling sub-routine call and return, instruction interpretation and execution.	3

3.	**Memory Subblock:** Memory organization; concepts of semi-conductor memory, CPU memory interaction, organization of memory modules, cache memory and related mapping and replacement policies, virtual memory	3
4.	**Pipelining:** Introduction to pipelining, Instruction pipeline, Arithmetic pipeline, Data hazards, instruction hazards, performance considerations.	2
5.	**I/O Subblock:** I/O techniques - interrupts, polling, DMA; Synchronous vs. Asynchronous I/O; Controllers. Introduction to VHDL concepts: examples to be taken up from the rest of the course for implementation.	3
Text Books		
1.	Computer Organization, Hamachar, Vranesic & Zaky.	
2.	Circuit Design with VHDL, Volnei Pedroni.	
References		
1.	Vincent. P. Heuring, Harry F. Jordan ―Computer System design and Architecture 2nd edition,Pearson, 2003.	
2.	Apman, Gabriele Jost, Ruud van van der Pas, ―Using OpenMP: Portable Shared Memory ParallelProgramming (Scientific and Engineering Computation), 1st edition, MIT Press, 2007.	
3.	H. J. Siegel.Interconnection Network for Large Scale Parallel Processing, McGraw Hill, 1990.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Design and Analysis of Algorithms	**Semester**	5^{th}	
Department	Computer Science & Engineering	**Course Code**	CST306	
Credits	04	**L**	**T**	**P**
Course Type	Theory	3	1	0

Course Objectives

- To understand asymptotic notations to analyze the performance of algorithms.
- To understand and apply various problem solving techniques such as divide and conquer, greedy algorithm, dynamic programming, etc.
- To solve given problem by selecting the appropriate algorithm design technique and justify the selection.
- To know the concepts of P, NP, NP-hard and NP-complete problems.

Learning Outcomes

This is a first course in algorithm design. Students will:

- Learn good principles of algorithm design;
- Learn how to analyze algorithms and estimate their worst-case and average-case behavior (in easy cases);
- Analyze the asymptotic performance of algorithms.
- Write rigorous correctness proofs for algorithms
- Become familiar with fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles;
- Learn how to apply their theoretical knowledge in practice (via the practical component of the course).

Course Synopsis

Basic strategies of algorithm design: top-down design, divide and conquer, average and worst-case criteria, asymptotic costs. Simple recurrence relations for asymptotic costs. Choice of appropriate data structures: arrays, lists, stacks, queues, trees, heaps, priority queues, graphs, hash tables. Applications to sorting and searching, matrix algorithms, shortest-path and spanning tree problems. Introduction to discrete optimisation algorithms: dynamic programming, greedy algorithms. Graph algorithms: depth first and breadth first search.

Course Outline / Content

Unit	Topics	Week
1.	**Analysis of Algorithms:** Algorithm Design paradigms, motivation. Review of algorithmic strategies, asymptotic analysis: upper and lower complexity bounds. Identifying differences among best, average and worst Case Behaviours. Big O, little O, omega and theta notations, Standard complexity classes. Empirical measurements of performance. Time and space trade-offs in algorithms. Analysing recursive algorithms using recurrence relations.	2
2.	**Divide & Conquer:** Structure of divide and conquer algorithms: examples, Binary search, Quick sort, analysis of divide and conquer run time recurrence relations.	2

	Greedy Algorithms: Overview of the greedy paradigm, examples of exact optimization solution (minimum cost spanning tree), approximate solution (Knapsack problem), single source shortest paths.	
3.	**Dynamic Programming**: Overview, difference between dynamic programming and divide and conquer, applications: shortest path in graph, matrix multiplication, travelling salesperson problem, longest common sequence.	2
4.	**Graph Algorithms:** Graphs and their Representations, Graph Traversal Techniques: Breadth First Search (BFS) and Depth First Search (DFS), Applications of BFS and DFS, Minimum Spanning Trees (MST), Prim's and Kruskal's algorithms for MST, Connected Components, Dijkstra's Algorithm for Single Source Shortest Paths, Floyd's Algorithm for All-Pairs Shortest Paths Problem.	3
5.	**Back Tracking:** Overview, 8-Queens problem and Knapsack problem. **Branch & Bound:** LC searching, bounding, FIFO branch and bound, Applications: 0/1 Knapsack problem, Travelling salesperson problem.	2
6.	**Computational complexity:** Complexity measures, Polynomial vs non-polynomial time complexity; NP hard and NP complete classes, Examples.	1
Text Books		
1.	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", PHI.	
2.	Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", Third Edition, Pearson Education, 2006	
3.	Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, 2011	
4.	Anany Levitin. "Introduction to the Design and Analysis of algorithms", Pearson.	
References		
1.	Steven S Skiena, "The Algorithm Design Manual" – Springer Publications	
2.	Knuth, " The Art of Programming", Addison Wesley Vol I and II	
3.	Michael T Goodrich, "Algorithm Design" WILEY Publications.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Microprocessor	**Semester**	5^{th}	
Department	Computer Science & Engineering	**Course Code**	CST307	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- To understand the basic Structure and Operations of microcomputer and
- To familiarize basic architecture of 8085 microprocessors and program 8085 Microprocessor using Assembly Level Language.
- To understand the System bus structure and its different operations.
- To understand interfacing of 16 bit microprocessor with memory and peripheral chips involving system design.
- To familiarize basic architecture of 8086 microprocessor

Learning Outcomes

- Write programs to run on 8085 microprocessor based systems.
- Design system using memory chips and peripheral chips for 16 bit 8085 microprocessor.
- Understand and devise techniques for faster execution of instructions, improve speed of operations and enhance performance of microprocessors.
- Distinguish between RISC and CISC processors.
- Understand multi core processor and its advantages.

Course Synopsis

Microcomputer Structure and Operations, Microprocessors and Memory, Assembly Language Programming, Bus System, Microprocessors Interfacing, Introduction to 8086 architecture.

Course Outline / Content

Unit	Topics	Week
1.	**Microcomputer Structure and Operations:** Basic Microcomputer Elements, Typical Microcomputer Structure, CPU, Memory System, Input Output	2
2.	**Microprocessors and Memory:** Typical 8, 16 and 32 bit Microprocessors,8085,Microprocessor Specification, Memory Technologies	2
3.	**Assembly Language Programming:** Programming Model of 8085, Registers, Fetch, Execute Operation of CPU, Instruction Set, Addressing Modes, Basic Operations, Microprocessor Arithmetic, Program Flow, Control Using Looping and Branching, Stack, Subroutines, Interrupts, Resets	3
4.	**Bus System:** System Bus Structure, Bus Operations, Cycle by Cycle Operations, Timing and Control, Priority Management, Address Decoding	2
5.	**Microprocessors Interfacing:** Interfacing concepts, Parallel Input Output, Memory Interfacing, Direct Memory Access. The Serial Subsystems. Programmable Peripheral Interface, Analog Converter	3

	Subsystem	
6.	**Introduction to 8086 architecture**: Main features and addressing modes, difference between 8085 and 8086.	2
Text Books		
1.	Microprocessor by Goankar	
2.	Microprocessor by Douglas Hall	
References		
1.	8086/8088 family: Design Programming and Interfacing: John Uffenbeck .	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Microprocessor Lab	**Semester**	5^{th}	
Department	Computer Science & Engineering	**Course Code**	CSL308	
Credits	01	**L**	**T**	**P**
Course Type	Lab	0	0	2

Course Objectives

- To become familiar with the architecture and Instruction set of Intel 8085 microprocessor.
- To expose students, to the operation of typical 8085 microprocessor trainer kit.
- To provide practical hands on experience with Assembly Language Programming.
- Develop and test assembly language programs to use instructions of 8085.
- Get familiarize with interfacing of various peripheral devices with 8085 using 8279 chip.

Learning Outcomes

By the end of this course, the students will be able to run programs on 8085 microprocessor based systems.

Course Synopsis

To enable a student to have a practical command over the concepts learned in the course.

Course Outline / Content

Unit	Topics	Week
1.	• Develop a program to add two double byte numbers. • Develop a subroutine to add two floating point quantities.	1
2.	Develop program to multiply two single byte unsigned numbers, giving a 16 bit product.	1
3.	Develop subroutine which will multiply two positive floating point numbers.	1
4.	Write program to evaluate P* Q*+R* & S are 8 bit binary numbers.	1
5.	Write a program to divide a 4 byte number by another 4 byte number	1
6.	Write a program to divide an 8 bit number by another 8 bit number up to a fractional quotient of 16 bit	1
7.	Write a program for adding first N natural numbers and store the results in memory location X.	1
8.	Write a program which decrements a hex number stored in register C. The Program should half when the program register reads zero.	1
9.	Write a program to introduce a time delay of 100 ms using this program as a subroutine display numbers from 01H to OAH with the above calculated time delay between every two numbers.	1
10.	N hex numbers are stored at consecutive memory locations starting from X. Find the largest number and store it at location Y.	1

11.	Interface a display circuit with the microprocessor either directly with the bus or by using I/O ports. Write a program by which the data stored in a RAM table is displayed.	1
12.	Design and interface a circuit to read data from an A/D converter, using the 8255 A in the memory mapped I/O.	1
13.	• Design and interface a circuit to convert digital data into analog signal using the 8255 A in the memory mapped I/O. • Interface a keyboard with the microprocessor using 8279 chip and transfer the output to the printer.	1
14.	Design a circuit to interface a memory chip with microprocessor with given memory map.	1
Text Books		
1.	Microprocessor by Goankar	
2.	Microprocessor by Douglas Hall	
References		
1.	8086/8088 family: Design Programming and Interfacing: John Uffenbeck	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Operating System	**Semester**	5^{th}	
Department	Computer Science & Engineering	**Course Code**	CST309	
Credits	04	**L**	**T**	**P**
Course Type	Theory	3	1	0

Course Objectives

- To understand the services provided by and the design of an operating system.
- To understand the structure and organization of the file system.
- To understand what a process is and how processes are synchronized and scheduled.
- To understand different approaches to memory management.
- Students should be able to use system calls for managing processes, memory and the file system.
- Students should understand the data structures and algorithms used to implement an OS.

Learning Outcomes

On completing this course the students should have acquired the following capabilities:

- An appreciation of the role of an operating system.
- Become aware of the issues in the management of resources like processor, memory and input-output.
- Should be able to select appropriate productivity enhancing tools or utilities for specific needs like filters or version control.
- Obtain some insight into the design of an operating system.

Course Synopsis

The course will provide an introduction to Operating Systems (OS), their design and implementation. We will discuss the goals of an OS, and some successful and not-so-successful OS designs. We will also discuss the following OS services in detail: thread scheduling, security, virtual memory, file system. In this course we will explore the core principles of operating systems design and implementation, including basic operating system structure; process and thread synchronization and concurrency; file systems and storage servers; memory management techniques; process scheduling and resource management; virtualization; and security.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction:** Operating system and function, Evolution of operating system, Batch, Interactive, Time Sharing and Real Time System, System protection.	1
2.	**Operating System Structure:** System Components, System structure, Operating System Services.	1
3.	**Concurrent Processes:** Process concept, Principle of Concurrency, Producer Consumer Problem, Critical Section problem, Semaphores, Classical problems in Concurrency, Inter Process Communication, Process Generation, Process Scheduling.	2
4.	**CPU Scheduling:** Scheduling Concept, Performance Criteria Scheduling Algorithm, Evolution, Multiprocessor Scheduling.	2
5.	**Deadlock:** System Model, Deadlock Characterization, Prevention, Avoidance and Detection, Recovery from deadlock	2

	combined approach	
6.	**Memory Management:** Base machine, Resident monitor, Multiprogramming with fixed partition, Multiprogramming with variable partition, Multiple base register, Paging, Segmentation, Virtual memory concept, Demand paging, Performance, Paged replaced algorithm, Allocation of frames, Thrashing, Cache memory, Organization, Impact on performance.	2
7.	**I/O Management & Disk Scheduling:** I/O devices and organization of I/O function, I/O Buffering, DISK I/O, Operating System Design Issues.	2
8.	**File System:** File Concept, File Organization and Access Mechanism, File Directories, File Sharing, Implementation Issues.	2
Text Books		
1.	J. Peterson, A. Silberschatz, and P. Galvin. Operating System Concepts, Addison Wesley, 3rd Edition, 1989.	
2.	Andrew S. Tannenbaum, "Modern Operating Systems", Prentice Hall.	
3.	William Stallings "Operating Systems – Internals and design principles", Prentice Hall	
References		
1.	D.M Dhamdhere: Operating systems - A concept based Approach, 3rd Edition, Tata McGraw- Hill, 2012.	
2.	P.C.P. Bhatt: Introduction to Operating Systems Concepts and Practice, 3rd Edition, PHI, 2010	
3.	Harvey M Deital: Operating systems, 3rd Edition, Pearson Education, 2011.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Python Programming	**Semester**	5^{th}	
Department	CSE	**Course Code**	CST310	
Credits	04	**L**	**T**	**P**
Course Type	Theory	3	1	0

Course Objectives

- To learn Syntax, Semantics and create Functions and to handle strings and files in Python.
- To understand Lists, Dictionaries, and Regular Expressions in Python.
- To implement OOP concepts in Python.
- To build web services and Introduction to Network and Database Programming in Python.

Learning Outcomes

The students should be able to:

- Understand Python syntax and semantics and be fluent in the use of Python flow control and functions.
- Demonstrate proficiency in handling Strings and File Systems.
- Implement Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.
- Interpret the concepts of Object-Oriented Programming as used in Python.
- Implement exemplary applications related to Network Programming, Web Services and Databases in Python.

Course Synopsis

Introduction to Python Programming; Building blocks of a python program: Variables, expressions and statements; Functions; Strings; Files; Lists; Dictionaries; Tuples; Regular expressions; Networked Programs; Unix Web Services; OOP; Using Databases

Course Outline / Content

Unit	Topics	Week
1.	Why should you learn to write programs, Variables, expressions and statements, Conditional execution, Functions	3
2.	Iteration, Strings, Files	3
3.	Lists, Dictionaries, Tuples, Regular Expressions	3
4.	Classes and objects, Classes and functions, Classes and methods	2
5.	Networked programs, Using Web Services, Using databases and SQL	3

Text Books

1.	Charles R. Severance, "Python for Everybody: Exploring Data Using Python 3", 1st Edition, CreateSpace Independent Publishing Platform, 2016. (http://do1.drchuck. com/pythonlearn/EN_us/pythonlearn.pdf) (Chapters 1 – 13, 15)
2.	Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", 2ndEdition, Green Tea Press, 2015. (http://greenteapress.com/thinkpython2/thinkpython2.pdf) (Chapters 15, 16, 17)

References

1.	Charles Dierbach, "Introduction to Computer Science Using Python", 1st Edition, Wiley India Pvt Ltd. ISBN-13: 978-8126556014
2.	Mark Lutz, "Programming Python", 4th Edition, O'Reilly Media, 2011.ISBN-13:

	978-9350232873
3.	Wesley J Chun, “Core Python Applications Programming”, 3rdEdition,Pearson Education India, 2015. ISBN-13: 978-9332555365
4.	Roberto Tamassia, Michael H Goldwasser, Michael T Goodrich, “Data Structures and Algorithms in Python”,1stEdition, Wiley India Pvt Ltd, 2016. ISBN-13: 978- 8126562176
5.	Roberto Tamassia, Michael H Goldwasser, Michael T Goodrich, “Data Structures and Algorithms in Python”,1stEdition, Wiley India Pvt Ltd, 2016. ISBN-13: 978- 8126562176

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Python Programming - Lab	**Semester**	5^{th}	
Department	Computer Science & Engineering	**Course Code**	CST311	
Credits	01	**L**	**T**	**P**
Course Type	Lab	0	0	2

Course Objectives

- To learn Syntax, Semantics and create Functions and to handle strings and files in Python.
- To understand Lists, Dictionaries, and Regular Expressions in Python.
- To implement OOP concepts in Python.
- To build web services and Introduction to Network and Database Programming in Python.

Learning Outcomes

The students should be able to:

- Understand Python syntax and semantics and be fluent in the use of Python flow control and functions.
- Demonstrate proficiency in handling Strings and File Systems.
- Implement Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.
- Interpret the concepts of Object-Oriented Programming as used in Python.
- Implement exemplary applications related to Network Programming, Web Services and Databases in Python.

Course Synopsis

Introduction to Python Programming; Building blocks of a python program: Variables, expressions and statements; Functions; Strings; Files; Lists; Dictionaries; Tuples; Regular expressions; Networked Programs; Unix Web Services; OOP; Using Databases

Course Outline / Content

Unit	Topics	Week
1.	Implement a sequential search	1
2.	Create a calculator program	1
3.	Explore string functions	1
4.	Implement Selection Sort	1
5.	Implement Stack	1
6.	Read and write into a file	1
7.	Demonstrate usage of basic regular expression	1
8.	Demonstrate use of advanced regular expressions for data validation.	1
9.	Demonstrate use of List	1
10.	Demonstrate use of Dictionaries	1
11.	Create Comma Separate Files (CSV), Load CSV files into internal Data Structure	1
12.	Write script to work like a SQL SELECT statement for internal Data Structure made in earlier exercise	1
13.	Write script to work like a SQL Inner Join for an internal Data Structure made in earlier exercise	1

Text Books

1.	Charles R. Severance, “Python for Everybody: Exploring Data Using Python 3”,

	1st Edition, CreateSpace Independent Publishing Platform, 2016. (http://do1.drchuck. com/pythonlearn/EN_us/pythonlearn.pdf) (Chapters 1 – 13, 15)
2.	Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", 2ndEdition, Green Tea Press, 2015. (http://greenteapress.com/thinkpython2/thinkpython2.pdf) (Chapters 15, 16, 17)
References	
1.	Charles Dierbach, "Introduction to Computer Science Using Python", 1st Edition, Wiley India Pvt Ltd. ISBN-13: 978-8126556014
2.	Mark Lutz, "Programming Python", 4th Edition, O'Reilly Media, 2011.ISBN-13: 978-9350232873
3.	Wesley J Chun, "Core Python Applications Programming", 3rdEdition,Pearson Education India, 2015. ISBN-13: 978-9332555365
4.	Roberto Tamassia, Michael H Goldwasser, Michael T Goodrich, "Data Structures and Algorithms in Python",1stEdition, Wiley India Pvt Ltd, 2016. ISBN-13: 978- 8126562176
5.	Roberto Tamassia, Michael H Goldwasser, Michael T Goodrich, "Data Structures and Algorithms in Python",1stEdition, Wiley India Pvt Ltd, 2016. ISBN-13: 978- 8126562176

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Data Communication	**Semester**	5^{th}	
Department	Computer Science & engineering	**Course Code**	CST312	
Credits	04	**L**	**T**	**P**
Course Type	Theory	3	1	0

Course Objectives

- To understand the concept of communication engineering, signals, channels and communication systems.
- To understand and analyze the characteristics of various modulation techniques.
- To critically analyze various modulation techniques used in modern communication systems
- To solve basic network design problems using knowledge of common local and wide area network architectures.
- To apply knowledge of computers, software, networking technologies and information assurance to an organization's management, operations, and requirements.

Learning Outcomes

By the end of this course, the student will be able to:

- Identify various components in a data communication system, describe their properties, explain how they work and evaluate their performance;
- Design solutions to solve engineering problems that require the applications of data and computer communication technology.

Course Synopsis

Sampling; Nyquist sampling theorem; Digital modulation Techniques; Line coding techniques; Data transmission; Multiplexing Techniques; Errors in data communication; Basic concept of network; LAN, MAN and WAN.

Course Outline / Content

Unit	Topics	Week
1.	**Data and Signals**: Data, Signals, Types of Signals, Bandwidth, spectrum, Digitization of analog signals, sampling, Nyquist sampling theorem, quantization, quantization noise, Pulse code modulation.	1
2.	**Digital modulation Techniques**: ASK, FSK, PSK, DPSK, M-ary PSK, QAM. Signal constellation.	1
3.	**Line coding techniques**: NRZ, RZ, Biphase, Manchester coding, AMI, HDBn.	2
4.	**Transmission media**: Guided and un-guided media, twisted wire pair, co-axial cable, optical fibre, microwave links, satellite microwave link, their characteristic features and applications for data transmission.	2
5.	**Data transmission**: simplex, half duplex and full duplex, Asynchronous and synchronous data transmission. Carrier, bit and frame synchronization techniques, Phase lock loop.	2
6.	**Multiplexing Techniques:** Frequency Division Multiplexing, Time Division Multiplexing, Wavelength division Multiplexing and Code Division Multiplexing. Spread Spectrum.	2
7.	**Errors in data communication**: Types of errors, error detection and correction techniques, forward error correction, polynomial error detection scheme, computation of CRC. Hardware.	2

8.	**Data communication network:** Basic concept of network, Advantages and applications, Types of networks (LAN, MAN and WAN), Different network topologies like star, ring, hybrid, tree.	2
Text Books		
1.	William Stallings: Data & Computer Communications, 7th Ed, PHI	
2.	Andrew Tanenbaum, "Computer Networks" PHI	
References		
1.	Sklar, "Digital Communications fundamentals & Applications"2nd Ed Pearson Pub	
2.	Keizer, " Local Area Networks" McGraw Hill	

National Institute of Technology Srinagar

6th Semester

Computer Science & Engineering

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Artificial Intelligence	**Semester**	6^{th}	
Department	Computer Science & Engineering	**Course Code**	CST352	
Credits	04	**L**	**T**	**P**
Course Type	Theory	3	1	0

Course Objectives

- To understand the fundamentals of computational intelligence
- To know about the various knowledge representation methods
- To understand the features of neural network and its implementation
- To study about various data clustering methods
- To gain knowledge in evolutionary computation and neuro – fuzzy systems

Learning Outcomes

- Implement computational intelligence through applications
- Understand knowledge representation methods and apply approximate reasoning
- Apply evolutionary algorithm to solve the optimization problem
- Gain research Knowledge to develop applications using hybrid systems
- Able to Model Flexible Fuzzy Inference systems for dynamic nonlinear data sets.

Course Synopsis

Introduction and history of AI; Knowledge representation; Inference mechanisms; Machine Learning and Expert systems.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction to Al And Production Systems** : Introduction to AI-Problem formulation, Problem Definition - Production systems, Control strategies, Search strategies. Problem characteristics, Production system characteristics - Specialized production system- Problem solving methods - Problem graphs, Matching, Indexing and Heuristic functions - Hill Climbing-Depth first and Breath first, Constraints satisfaction - Related algorithms, Measure of performance and analysis of search algorithms.	2
2.	**Representation of Knowledge**: Game playing - Knowledge representation, Knowledge representation using Predicate logic, Introduction to predicate calculus, Resolution, use of predicate calculus, Knowledge representation using other logic-Structured representation of knowledge.	3
3.	**Knowledge Inference**: Knowledge representation -Production based system, Frame based system. Inference - Backward chaining, Forward chaining, Rule value approach, Fuzzy reasoning - Certainty factors, Bayesian Theory-Bayesian Network-Dempster - Shafer theory.	3
4.	**Planning and Machine Learning**: Basic plan generation systems - Strips -Advanced plan generation systems – K strips - Strategic explanations -Why, Why not and how explanations.	3

	Learning- Machine learning, adaptive Learning.	
5.	**Expert Systems**: Expert systems - Architecture of expert systems, Roles of expert systems - Knowledge Acquisition – Meta knowledge, Heuristics. Typical expert systems - MYCIN, DART, XOON, Expert systems shells.	3
Text Books		
1.	Deepak Khemani. A First Course in Artificial Intelligence, McGraw Hill Education (India), 2013	
2.	Stuart Russell and Peter Norvig.ArtificialIntelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2009.	
References		
1.	Stefan Edelkamp and Stefan Schroedl. Heuristic Search: Theory and Applications, Morgan Kaufmann, 2011.	
2.	John Haugeland,Artificial Intelligence: The Very Idea, A Bradford Book, The MIT Press, 1985	
3.	Pamela McCorduck, Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence, A K Peters/CRC Press; 2 edition, 2004.	
4.	Zbigniew Michalewicz and David B. Fogel. How to Solve It: Modern Heuristics. Springer; 2nd edition, 2004.	
5.	Judea Pearl. Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, 1984.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Artificial Intelligence Lab	**Semester**	6^{th}	
Department	CSE	**Course Code**	CSL353	
Credits	01	**L**	**T**	**P**
Course Type	Lab	0	0	2

Course Objectives

- To understand the concepts of Artificial intelligence and machine learning.
- To understand and practice prolog and python.
- To understand and practice various classifiers like SVM and neural networks.
- To understand and practice logic and reasoning through logic programming.

Learning Outcomes

- Develop the basic logic programs.
- Understanding the implementation of various machine learning libraries like torch, tensor flow.
- Hands on experience on applying various classifiers on MNIST dataset.

Course Outline / Content

Unit	Topics	Week
1.	Study of PROLOG.	1
2.	Write a program to solve 8 queens' problem	1
3.	Solve any problem using depth first search.	1
4.	Solve any problem using best first search.	1
5.	Solve 8-puzzle problem using best first search	2
6.	Solve Robot (traversal) problem using means End Analysis	2
7.	Solve traveling salesman problem.	2
8.	Implementation of Linear and Logistic regression.	2
9.	Implementing classifiers on MNIST Data set.	2

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Computer Networks	**Semester**	6th	
Department	Computer Science &Engineering	**Course Code**	CST354	
Credits	04	**L**	**T**	**P**
Course Type	Theory	3	1	0

Course Objectives

- Understand network models and architectures.
- Specifyandidentifydeficienciesinexistingprotocols,andthengoontoformulatenewand betterprotocols.
- Analyse, specify and design the topological and routing strategies for an IP based networkinginfrastructure.
- Explain concepts and theories of networking and apply them to various situations, classifying networks

Learning Outcomes

Upon completion of the course the student will be able to:

- Describe and analyse the hardware, software, components of a network and the interrelations.
- Explain networking protocols and their hierarchical relationship
- Compare protocol models and select appropriate protocols for a particular design.
- Develop solutions for networking
- Explain concepts and theories of networking and apply them to various situations, classifying networks

Course Synopsis

The course aims to familiarise the student with networking concepts, protocols and the internet.

Course Outline / Content

Unit	Topics	Week
1.	**Basic concept of network:** Advantages and applications, Types of networks (LAN, MAN and WAN), Different network topologies like star, ring, hybrid, tree.	1
2.	**Network Protocol Architecture**: OSI Reference model, Layers of the OSI model. Physical, Data-link, Network, Transport, Session, Presentation and Application layer.	1
3.	**Network Switching Techniques:** Circuit switched, message switching and packet switched networks, Datagram and virtual circuit services, Frame relay, ATM	2
4.	**Flow and Error Control:** Stop and wait flow control, Sliding window flow control, error control protocols, ARQ techniques, Stop-&-wait ARQ, Go back by N ARQ, Selective repeat ARQ.	2
5.	**Routing algorithms:** Routing tables, features of a routing algorithm, classification, optimality principle, sink tree, shortest path algorithm, Dijkstra algorithm, flooding, fixed routing, random routing, adaptive routing, distance vector and link state algorithm. **Congestion Control:** Congestion in networks and quality of	2

	service.	
6.	**Medium Access Control Protocols**: TDMA, FDMA, CDMA, ALOHA, Slotted ALOHA, CSMA, CSMA/CD, Ethernet, Token Ring network	2
7.	**Network security:** Need for network data security, plaintext, cyphertext, encryption techniques, substitution, transposition, DES encryption standard, Private key, public key, Authentication.	2
8.	**Internetworking and Internet fundamentals:** Network Interconnections, Bridges, Routers, Internet Concepts, Brief concepts about common Channel signalling and Integrated DigitalNetworking.	2
Text Books		
1.	William Stallings: Data & Computer Communications, 7th Ed, PHI	
2.	Andrew Tanenbaum, ―Computer Networks, PHI	
3.	Peterson and Davie, "Computer Networks, A Systems Approach", 5th ed., Elsevier, 2011	
References		
1.	Keizer, ― Local Area Networks, McGraw Hill	
2.	Sklar, ―Digital Communications fundamentals & Applications‖ 2nd Ed Pearson Pub.	
3.	Ying-Dar Liu, Ren-Hung Hwang, Fred Baker, "Computer Networks: An Open Source Approach", McGraw-Hill, 2011.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Computer Networks Lab	**Semester**		6th
Department	Computer Science & Engineering	**Course Code**		CSL355
Credits	01	**L**	**T**	**P**
Course Type	Lab	0	0	2

Course Objectives

- To gain a firm understanding of networking concepts learned in the course work by practical demonstration.
- Have working knowledge of the protocols to be used at various levels of the architecture

Learning Outcomes

The student should be able to:

- Set up LAN for home, office and similar configurations
- Secure the network by installing of firewalls and other security measures.

Course Synopsis

The objective of the lab course to familiarise students with networking concepts from a practical perspective.

Course Outline / Content

Unit	Topics	Week
1.	Introduction & Network Wire Crimping	1
2.	Ethernet	1
3.	Token Ring	1
4.	Switched LANs	1
5.	Network Design	1
6.	ATM	1
7.	RIP: Routing Information Protocol	1
8.	OSPF: Open Shortest Path First	1
9.	TCP: Transmission Control Protocol	1
10.	Queuing Disciplines	1
11.	RSVP: Resource Reservation Protocol	1
12	Firewalls and VPN	1
1.	Applications	2

Text Books

1.	William Stallings: Data & Computer Communications, 7th Ed, PHI
2.	Andrew Tanenbaum, ―Computer Networks, PHI
3.	Peterson and Davie, "Computer Networks, A Systems Approach", 5th ed., Elsevier, 2011

References

1.	Keizer, ― Local Area Networks, McGraw Hill
2.	Sklar, ―Digital Communications fundamentals & Applications‖ 2nd Ed Pearson Pub.
3.	Ying-Dar Liu, Ren-Hung Hwang, Fred Baker, "Computer Networks: An Open Source Approach", McGraw-Hill, 2011.

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Theory of Computation	**Semester**	6^{th}	
Department	Computer Science & Engineering	**Course Code**	CST356	
Credits	04	**L**	**T**	**P**
Course Type	Theory	3	1	0

Course Objectives

- Classify machines by their power to recognizelanguages.
- Employ finite state machines to solve problems incomputing.
- To design grammars and recognizers for different formal languages
- Comprehend the hierarchy of problems arising in the computersciences.

Learning Outcomes

After completing this course, the student should be able to:

- Understand various Computing models like Finite State Machine, Pushdown Automata, and Turing Machine;
- Understand Decidability and Undecidability of various problems.

Course Synopsis

Complexity of computations; theorems and proofs; Finite Automata; context free grammar; pushdown automata; concepts in parsing; Turing machines; Complexity theory.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction:** Complexity of computations, automata, computability, complexity, mathematical notions and terminology, definitions, theorems and proofs, types of proofs.	3
2.	**Automata & Languages:** Finite Automata, Non-determinism, regular expressions, non-regular expressions	3
3.	**Context free languages:** context free grammar, pushdown automata, non-context free languages, equivalences, closure properties, concepts in parsing.	3
4.	**Computability theory:** Turing machines, variants of Turing machines, the definition of Algorithm, Decidability, reducibility, advanced topics in computability theory- recursion theorem etc.	3
5.	**Complexity theory-** time complexity, space complexity, intractability.	2

Text Books

1.	C. Papadimitrou and C. L. Lewis. Elements of Theory of Computation, Prentice-Hall,1981.
2.	J.E. Hopcroft and J.D. Ullman. Introduction to Antomata Theory

References

1.	Languages of Computations, Addison-Wesley, 1979. (Indian edition available from Narosa.)
2.	Michael Sipser, “Theory of Computation” , Cengage Learning.

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Computer Graphics	**Semester**	6th	
Department	Computer Science & Engineering	**Course Code**	CST357	
Credits	04	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- Students will demonstrate an understanding of contemporary graphics hardware and of basic terminology, scope and techniques of Computer Graphics.
- Demonstrate and implement the 2D primitive drawing algorithms
- Demonstrate area filling algorithms, line/ Polygon clipping along with various 2D transformations2D viewing and Coordinate representations.
- Understand the 3D graphic primitives along with various Transformations and Other algorithmsand Projection Techniques for representing 3D graphic objects

Learning Outcomes

- Students will have an understanding of 2D graphics and algorithms including: line drawing, polygon filling, clipping, and transformations.
- Students will understand the techniques used in 3D computer graphics, including viewing transformations, hierarchical modelling, colour, lighting and texture mapping.

Course Synopsis

Introduction to graphics primitives; geometric transformation; parallel and perspective projection; Shading models; picture synthesis and analysis.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction**: Co-ordinate representation, Pixel, Raster Scan & Random Scan methods, colour CRT Raster scan basics, video basics, interactive devices, graphics input and output devices, mouse, track ball, light pen, digitizer, thumb wheel, raster scan graphics.	2
2.	**Graphics Primitives**: 2D Primitives - Output primitives – Line, Circle and Ellipse drawing algorithms - Attributes of output primitives – Two dimensional Geometric transformation - Two dimensional viewing –Line, Polygon, Curve and Text clipping algorithms.	3
3.	**Parallel and Perspective projections**:Three dimensional object representation –Polygons, Curved lines, Splines, Quadric Surfaces- Visualization of data sets - 3D transformations – Viewing -Visible surface identification. Basic Raster Graphics Algorithms. Geometric Modelling in 3-D. Viewing in 3-D. Concept of Synthetic Camera. Dialogue Design. Graphics User Interfaces. Windowing Systems.	3
4.	**Rendering**: Introduction to Shading models – Flat and Smooth shading – Adding texture to faces – Adding shadows of objects – Building a camera in a program – Creating shaded objects – Rendering texture – Drawing Shadows, Graphical Modelling of	3

	Discrete events.	
5.	**Introduction to Picture Synthesis and Analysis**: Conceptual Framework of an Interactive Graphical Simulation System. Simulation of Discrete Event Displays, Animation Techniques, Basic Rules for Animation. Graphical Simulation of continuous motion. Role of Virtual Reality in Graphical Simulation.	3
Text Books		
1.	Computer Graphics by Hearn and Baker, PHI	
2.	Preparata, Shamos, Computational Geometry- An Introduction.	
References		
1.	Procedural Elements for Computer Graphics by Rogers, TMH.	
2.	Mathematical Elements for Computer Graphics by Rogers and Adams, Mac Graw Hills	
3.	Computer Graphics: Schaum's Outline of Computer Graphics by Roy A Plastock	
1.	Research papers/Journal Articles from Standard Sources	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Computer Graphics Lab	**Semester**	6^{th}	
Department	Computer Science & Engineering	**Course Code**	CSL358	
Credits	01	**L**	**T**	**P**
Course Type	Lab	0	0	2

Course Objectives

- Implement the 2D primitive drawing algorithms
- Demonstrate and Implement the 2D transformation techniques
- Demonstrate and implement the 3D transformation techniques
- Implement Animation scenes

Learning Outcomes

By the end of this course, the students will beable to:

- Implement the algorithms for drawing the basic graphic primitives.
- Apply different kinds of transformations.
- Draw three dimensional objects.
- Generate fractal images.

Course Synopsis

Bresenham's algorithms for drawing line, circle and ellipse; Two dimensional transformations, Three dimensional transformations, Composite transformations.

Course Outline / Content

Unit	Topics	Week
1.	Implementation of Bresenham's Algorithm – Line, Circle, Ellipse. Implementation of Line, Circle and ellipse Attributes.	2
2.	Two Dimensional transformations - Translation, Rotation, Scaling, Reflection, Shear.	2
3.	Composite 2D Transformations.	2
4.	Cohen Sutherland 2D line clipping and Windowing	1
5.	Sutherland – Hodgeman Polygon clipping Algorithm.	1
6.	Three dimensional transformations - Translation, Rotation, Scaling.	2
7.	Composite 3D transformations.	2
8.	Drawing three dimensional objects and Scenes.	1
9.	Generating Fractal images.	1

Text Books

1.	Computer Graphics by Hearn and Baker, PHI
2.	Preparata, Shamos, Computational Geometry- An Introduction.

References

1.	Procedural Elements for Computer Graphics by Rogers, TMH.
2.	Mathematical Elements for Computer Graphics by Rogers and Adams, Mac Graw Hills.
3.	Computer Graphics: Schaum's Outline of Computer Graphics by Roy A Plastock.
4.	Research papers/Journal Articles from Standard Sources.

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	JAVA Programming	**Semester**	6^{th}	
Department	Computer Science & Engineering	**Course Code**	CST359	
Credits	03	**L**	**T**	**P**
Course Type	Theory	2	0	2

Course Objectives

- The Students will learn to create Classes and their Objects.
- Learn and implement principles and concepts of Object Orientation such as Abstraction, Data Hiding, Polymorphism.
- Develop programs by using inbuilt libraries and importing Packages. The students will learn to create and handle threads, interfaces and applets.

Course Synopsis

On completion of the course the student should be able to: Use an integrated development environment to write, compile, run, and test simple object-oriented **Java** programs. Read and make elementary modifications to Java programs that solve real-world problems.

Course Outline / Content

Unit	Topics	Week
1.	**Overview of Basic OOP Concepts:** Need for object-oriented paradigm: Agents, responsibility, messages, methods, classes and instances, class hierarchies (Inheritance), method binding, datatypes, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, classes and objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, string handling, inheritance, super keyword, polymorphism- method overriding, abstract classes.	2
2.	**Packages and Interfaces:** Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring packages – Java.io, Java.util.	2
3.	**Exception handling and multithreading:** Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception sub classes. Differences between multi threading and multitasking, thread life cycle, creating threads, synchronizing threads, daemon threads, thread groups.	2
4.	**Event Handling:** Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes, inner classes.The AWT class	1

	hierarchy, user interface components- labels, button, canvas, scrollbars, text components, check box, check box groups, choices, lists panels – scrollpane, dialogs, menubar, graphics, layout manager – layout manager types – boarder, grid, flow, card and grib bag.	
5.	**Applets:** Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets.	2
6.	**Swing:** Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing-JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons – The JButton class, Check boxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees, and Tables.	2
7.	**JDBC**: JDBC Drivers, JDBC API,Executing statements, prepared statements and callable statements	1
Text Books		
1.	An Introduction to programming and OO design using Java, J.Nino and F.A. Hosch, John wiley& sons.	
2.	An Introduction to OOP, second edition, T. Budd, pearson education.	
References		
1.	Introduction to Java programming 6th edition, Y. Daniel Liang, Pearson education.	
2.	An introduction to Java programming and object oriented application development, R.A. Johnson-Thomson.	
3.	. Core Java 2, Vol 1, Fundamentals, Cay.S.Horstmann and Gary Cornell, seventh Edition, Pearson Education.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Practical Training & Tour	**Semester**		6^{th}
Department	Computer Science & Engineering	**Course Code**		CSI360
Credits	01	**L**	**T**	**P**
Course Type	Lab	0	0	2

Course Objectives

Practical training - is defined as an integrative hands-on learning experience in a supervised setting aimed at the professional preparation and training of a student. Students should be exposed to various areas of the organization in which they work. Practical training provides learning opportunities related to all parts of the curriculum. The student always works with the support and appropriate help from the field instructor. However, the student is engaged in carrying out a particular activity, so the responsibility is the student's.

Field courses (practical trainings) take place away from the College, usually at an industry, research institution or any other company.

Learning Outcomes

The overall goal of the practical training is:

- To get the field experience, to deepen professional education,
- To bring the theory to life (apply the knowledge, concepts and skills in a real working environment)
- To provide students with practice experience out of college, in organizational setting.
- To gain additional insight into the realistic work situations
- To apply knowledge and skill in practice
- To integrate classroom experience with work experience
- To increase the student´s professional self-awareness
- To gain practical work experience
- To complement the knowledge and skills learned in classes
- To provide opportunity to apply the knowledge and the skills in a practice-based setting
- To assist / carry out real tasks and duties
- Provide career guidance to the students
- To allow students to participate in practical lab work, meetings, conferences, trainings or other learning opportunities
- To experience responsible interaction with professionals
- To get feedback from field to class.

National Institute of Technology Srinagar

7th Semester

Computer Science & Engineering

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Compiler Design	**Semester**	7^{th}	
Department	Computer Science & Engineering	**Course Code**	CST415	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	1	0

Course Objectives

The objective of this course is to provide To understand different phases of compilation process.

- To analyse and implement various parsing techniques.
- To understand and analyse intermediate code.
- To realize the importance of code optimization and code generation

Learning Outcomes

After completion of this course the students should be able to have a basic understanding required for design of compilers.

Course Synopsis

Structure of compiler, Lexical analysis, Syntax analysis, Bottom up and top down evaluation of attributes, type checking, storage organization, intermediate code generation, basic blocks and flow graphs, register allocation, code generation, peep hole optimization, code generator generators.

Course Outline / Content

Unit	Topics	Week
1.	**Compiler structure:** analysis-synthesis model of compilation, various phases of a compiler, tool based approach to compiler construction.	1
2.	**Lexical analysis:** Interface with input, parser and symbol table, token, lexeme and patterns. Difficulties in lexical analysis, Error	1

	reporting and Implementation. Regular definition, Transition diagrams, LEX.	
3.	**Syntax analysis:** CFGs, ambiguity, associativity, precedence, top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing, bottom up parsing, operator precedence grammars, LR parsers (SLR, LALR, LR), YACC.	2
4.	**Syntax directed definitions**: inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top down evaluation of attributes, L- and S-attributed definitions.	2
5.	**Type checking:** type system, type expressions, structural and name equivalence of types, type conversion, overloaded functions and operators, polymorphic functions.	2
6.	**Run time system:** storage organization, activation tree, activation record, parameter passing, symbol table, dynamic storage allocation.	2
7.	**Intermediate code generation:** intermediate representations, translation of declarations, assignments, control flow, Boolean expressions and procedure calls and Implementation issues.	2
8.	**Code generation and instruction selection:** issues, basic blocks and flow graphs, register allocation, code generation, dag representation of programs, code generation from DAGs, peep hole optimization, code generator generators, specifications of machine.	2
Text Books		
1.	A. V. Aho, R. Sethi, and J. D. Ullman. Compilers: Principles, Techniques and Tools, Addison-Wesley, 1988.	
2.	C. Fischer and R. LeBlanc. Crafting a Compiler, Benjamin Cummings, 1991.	
References		
1.	A. C. Holub. Compiler Design in C, Prentice-Hall Inc., 1993. Appel. Modern Compiler Implementation in C: Basic Design, Cambridge Press.	
2.	Fraser and Hanson. A Retargetable C Compiler: Design and Implementation , Addison-Wesley.	

Department of Computer Science & Engineering

National Institute of Technology Srinagar

Course Title	Compiler Design Lab	**Semester**	7^{th}	
Department	Computer Science & Engineering	**Course Code**	CSL416	
Credits	01	**L**	**T**	**P**
Course Type	Lab	0	0	2

Course Objectives

- To understand the implementation of lexical analyser, parser and other compiler design aspects.
- To write codes for various top-down and bottom-up parsers and verify them for correctness.
- To understand Linux Utility Lex and Yacc tools.

Learning Outcomes

After completing this course the students should be able to understand the compiler coding and working in detail.

Course Synopsis

The Lab intends to make students implement lexical analysers and code for each of the following phases of a compiler:

- Syntax Analysis
- Semantic Analysis
- Intermediate Code Generation
- Code Optimization
- Code Generation

Course Outline / Content

Unit	Topics	Week
1.	Design a lexical analyzer for given language and the lexical analyzer should ignore redundant spaces, tabs and new lines.	2

2.	Simulate First and Follow of a Grammar	2
3.	Develop an operator precedence parser for a given language.	1
4.	Construct a recursive descent parser for an expression.	1
5.	Construct a LL(1) parser for an expression	2
6.	Design predictive parser for the given language	1
7.	Implementation of shift reduce parsing algorithm.	1
8.	Design a LALR bottom up parser for the given language.	1
9.	Implement the lexical analyzer using JLex, flex or lex or other lexical analyzer generating tools	1
Text Books		
1.	A. V. Aho, R. Sethi, and J. D. Ullman. Compilers: Principles, Techniques and Tools , Addison-Wesley, 1988.	
2.	C. Fischer and R. LeBlanc. Crafting a Compiler, Benjamin Cummings, 1991.	
References		
1.	A. C. Holub. Compiler Design in C, Prentice-Hall Inc., 1993. Appel. Modern Compiler Implementation in C: Basic Design, Cambridge Press.	
2.	Fraser and Hanson. A Retargetable C Compiler: Design and Implementation, Addison-Wesley.	

Department of Computer Science & Engineering

National Institute of Technology Srinagar

Course Title	Network Security	**Semester**	7^{th}	
Department	Computer Science & Engineering	**Course Code**	CST417	
Credits	04	**L**	**T**	**P**
Course Type	Theory	3	1	0

Course Objectives

- To help the students to understand important security goals in the networks- Confidentiality, Integrity, Authenticity, Non-repudiation and Availability and cryptographic techniques to implement these security goals.
- To provide a necessary review of mathematical concepts to implement different cryptographic techniques to achieve the network security goals and then provides a deeper dive to the field of cryptography- symmetric and asymmetric key cryptography and methods to implement them.
- The course provides a top down approach to explore the security implementations in different network layers-application, transport and network.

Learning Outcomes

After completion of this course the students should be able to:

- Visualize the security goals clearly in the networks.
- Analyse the basic concepts of network security to predict and classify attacks on a system/network.
- Understand and apply authentication techniques to provide secure communication.
- Assess the security threats to ICT infrastructure using modern tools such as firewalls, UTMs, etc.

Course Synopsis

Cryptography, classical encryption, Divisibility, Modular Arithmetic,Random Numbers.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction** Review of Layered Architecture of the Network - the OSI Reference Model, Computer Security Concepts, The OSI Security Architecture, Security - Attacks, Services and Mechanism**s.**	1
2.	**Cryptography Introduction: Classical Encryption Techniques** Techniques for Implementing Security Goals: An Overview of Cryptography and Steganography, A brief recap of cryptographic principles and motivations for secure network, General thoughts on breaking the cryptosystems.	1
3.	**Review & Self Study - Mathematics for Symmetric Key Cryptography** Review of important mathematical concepts: Divisibility, Modular Arithmetic, Groups and Rings. Classical Encryption Techniques - Substitution and Transposition	2
4.	**Random Number (PRN) Generation and Stream Cipher** Random Numbers, True Random Number Generators, Pseudo Random Numbers - principles and generators, Cryptographically Secure Random Number Generators, One Time Pad, Stream Cipher- RC4	2
5.	**Block Ciphers - Data Encryption Standard (DES) and Advanced Encryption Standard (AES)** Block Cipher Structure, Introduction to Data Encryption Standard, Triple DES - introduction, structure & implementation, Fields, Finite Fields - GF(p), GF(2n) and polynomial arithmetic. Advanced Encryption Standard (AES) - Introduction, structure & implementation.	2
6.	**Block Cipher Operations** Modes of Operations, Electronic Code Book Mode, Cipher Block Chaining Mode, Output Feedback Mode, Cipher Feedback Mode, Counter Mode, Enhancing the Security of Block Ciphers: Multiple Encryption, 3DES and DESX, Meet-in-the Middle	2

	Attack.	
7.	**Data Integrity** Introduction & Motivation, Hash Functions from Block Cipher, Message Digest (MD) Hash Family, Secure Hash Algorithm (SHA-1 and SHA-3), Message Authentication Codes (MAC).	2
8.	**Review & Self Study - Mathematics for Asymmetric Key Cryptography** Review of important mathematical concepts used in asymmetric key cryptography – Euclidean and Extended Euclidean Algorithm, Euler's Phi Function, Prime Numbers and Primality Testing, Euler's and Fermat's Theorem.	2
9.	**Asymmetric Key Cryptography** Introduction & Principles of Asymmetric Key Cryptography, Different Public Key Algorithms, Introduction to RSA, RSA in Practice and Attacks, Diffe-Hellman Key Exchange.	1
10.	**Digital Signatures** Introduction & Motivation, Principles and Applications, RSA based Digital Signature, RSA Probilistic Signature Scheme.	1
11.	**Mutual Trust - Key Management and User Authentication** Introduction and challenges in key distribution, Symmetric Key Distribution and Agreement, Public Key Distribution, Principles of user authentication, User Authentication Protocol – Kerberos, Public-Key Infrastructure.	1
12.	**Security at the Application Layer** Application Layer Security - Objectives, Issues and Need, Email Security, Pretty Good Privacy, Secure/Mulitpurpose Internet Mail Extension, Domain Keys Identfied Mail.	1
13.	**Security at the Transport Layer** Web Security: Threats and Challenges, Securing Web-based transactions at the transport layer, Secure Socket Layer, Transport Layer Security, HTTPS, Combining HTTP and SSL/TLS - the secure HTTPS, Remote login, Challenges, One Possible Solution Approach – SSH, Wireless Security – TLS and WAP End-to-End Security.	1

14.	**Security at the Network Layer** IP Security: Overview and Policy, Encapsulating Security Payload, Combining Security Associations, Internet Key Exchange, Cryptogaphic Suites.	1
15.	**System Security** Malwares – Virus, Worms, etc, Malicious Software and Anti-malwares, Distributed Denial of Service Attacks, Intruders and Intrusion Detection, Firewall Need and Characteristics, Types of Firewalls and Biasing, Firewall Location and Confgurations.	2

Text Books

1.	Stallings William: Cryptography and Network Security - Principles and Practice, Pearson India, 6th Edition, 2014.

References

1.	ChristofPaar and Jan Pelzl: Understanding Cryptography - A Textbook for Students and Practitioners, Springer, 1st Edition, 2010.
2.	SchneierBruice: Applied Cryptography : Protocols, Algorithms And Source Code In C, Wiley India, 2nd Edition, Reprint - 2013.
3.	Kurose James F and Keith W. Ross: Computer Networking: A Top-Down Approach, Pearson India, 5th Edition, 2012.

Department of Computer Science & Engineering

National Institute of Technology Srinagar

Course Title	Network Security Lab	**Semester**	7^{th}	
Department	Computer Science & Engineering	**Course Code**	CSL418	
Credits	01	**L**	**T**	**P**
Course Type	Lab	0	0	2

Course Objectives

- To understand principles of web security and to guarantee a secure network by monitoring and analysing the nature of attacks through cyber/computer forensics software/tools
- Exhibit knowledge to secure corrupted systems, protect personal data, and secure computer networks in an Organization.
- To have the ability to compare merits and demerits of different Cryptographic techniques and take decisions while securing a network

Learning Outcomes

At the end of the Lab, students should be able to :

- Analyse and evaluate the cyber security needs of an organization.
- Determine and analyse software vulnerabilities and security solutions to reduce the
- risk of exploitation.
- Measure the performance and troubleshoot cyber security systems

Course Synopsis

Implementation of Cryptography; Threats and Vulnerabilities; Understand the Tools and Techniques.

Course Outline / Content

Unit	Topics	Week
1.	Setting Up the System for testing purpose: Learning Basic Commands.	1

2.	Software Requirements. Security Attacks : ARP Attacks (ARP Cache Poisoning, ARP Man in the Middle Attack).	2
3.	IP Attacks (IP Fragmentation Attack, IP Teardrop Attack)	2
4.	ICMP Attacks(Ping of Death, Smurf Attack, ICMP Destination Unreachable, ICMP Redirect, ICMP Source Quench)	2
5.	TCP Attacks (SYN Flooding Attack, TCP RST Attack), UDP Attack.	3
6.	Understand the Tools and Techniques: IEXPRESS 2.0, CAY KARAT, Damm Web Application Vulnerabilities (DWAV), WebGoat, ProRat Trojan, Key Logger, Steganographer etc.	3
7.	Nmap Port Scanning: TCP Port Scanning (TCP Connect () Scanning, TCP SYN Scan, TCP FIN Scan, XMAS Scan, TCP NULL Scan)	3
8.	UDP Port Scanning, Performing Stealth Scan of a Selected Computer.	2
Books		
1.	SchneierBruice: Applied Cryptography : Protocols, Algorithms And Source Code In C, Wiley India, 2nd Edition, Reprint - 2013.	

Department of Computer Science & Engineering

National Institute of Technology Srinagar

Course Title	Pre-Project	**Semester**	7^{th}	
Department	Computer Science & Engineering	**Course Code**	CSP419	
Credits	03	**L**	**T**	**P**
Course Type	Practical	0	0	4

Course Objectives

- To enhance the student's knowledge and skills in solving problem through structured project research in order to produce a competent and productive engineer.

Learning Outcomes

Upon completion of Pre-Project, student should be able to:

- Identify and describe the problem and scope of project clearly.
- Collect, analyze and present data into meaningful information using relevant tools.
- Select, plan and execute a proper methodology in problem solving.
- Work independently and ethically.
- Present the results in written and oral format effectively.
- Identify basic entrepreneurship skills in project management

Course Synopsis

Final Year Project (FYP) is the individual project, which takes place over the two semesters. It is a mandatory subject for students to be awarded with Bachelor Degree. In the FYP, students are expected to undergo research studies which relate to the major course offered in the faculty.

The final year project consists of two parts; the first part FYP 1 is a prerequisite to the second part, FYP 2.

These two parts of FYP are under subject code CSE-705 and CSE-801 respectively.

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Seminar	**Semester**	7^{th}	
Department	Computer Science & Engineering	**Course Code**	CSS420	
Credits	01	**L**	**T**	**P**
Course Type	Theory	0	0	2

Course Objectives

Each and every student has to deliver a seminar on recent research/technical topics related to CSE through power point presentations. Duration of presentation should be minimum of 30 minutes. Detailed breakup of marks to be decided by the course instructor depending on quality of content, presentation, answer to queries, communication skills etc.

Learning Outcomes

To promote further development of student success skills, such as reading and speaking; help students gain intellectual confidence; build in the expectation of academic success; have insight into latest research topics.

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Internet & Web Design	**Semester**	MTech	
Department	Computer Science & Engineering	**Course Code**	CSL501	
Credits	02	**L**	**T**	**P**
Course Type	Theory	1	0	2

Course Objectives

1. The aim of this course is to provide you the conceptual and technological developments in the field of Internet and web designing with the emphasis on comprehensive knowledge of Internet,
2. Its applications and the TCP/IP protocols widely deployed to provide Internet connective worldwide.
3. The World Wide Web with its widespread usefulness has become an integral part of the Internet. Therefore, this course also puts emphasis on basic concepts of web design.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction to Internet** Internet, Growth of Internet, Owners of the Internet, Anatomy of Internet, ARPANET and Internet history of the World Wide Web, basic Internet Terminology, Net etiquette. Internet Applications – Commerce on the Internet, Governance on the Internet, Impact of Internet on Society – Crime on/through the Internet.	3
2.	**TCP/IP – Internet Technology and Protocol** Packet switching technology, Internet Protocols: TCP/IP, Router, Internet Addressing Scheme: Machine Addressing (IP address), E-mail Addresses, Resources Addresses	1
3.	**Internet Connectivity** Connectivity types: level one, level two and level three connectivity, Setting up a connection: hardware requirement, selection of a modem, software requirement, modem configuration, Internet accounts by ISP: Telephone line options, Protocol options, Service options, Telephone line options – Dialup connections through the telephone system, dedicated connections through the telephone system, ISDN, Protocol options – Shell, SLIP, PPP, Service options – E-mail, WWW, News Firewall etc.	2
4.	**Internet Network** Network definition, Common terminologies: LAN, WAN, Node, Host, Workstation, bandwidth, Interoperability, Network administrator, network security, Network Components: Severs, Clients, Communication Media, Types of network: Peer to Peer, Clients Server, Addressing in Internet: DNS, Domain Name and their organization, understanding the Internet Protocol Address. Network topologies: Bust,	2

	star and ring, Ethernet, FDDI, ATM and Intranet. Services on Internet (Definition and Functions) 04 Hrs.	
5.	**Electronic Mail** Email Networks and Servers, Email protocols –SMTP, POP3, IMAp4, MIME6, Structure of an Email – Email Address, Email Header, Body and Attachments, Email Clients: Netscape mail Clients, Outlook Express, Web based E-mail. Email encryption- Address Book, Signature File.	1
6.	**Current Trends on Internet** Languages, Internet Phone, Internet Video, collaborative computing, e-commerce. Web Publishing and Browsing 10 Hrs. Overview, SGML, Web hosting, HTML. CGL, Documents Interchange Standards, Components of Web Publishing, Document management, Web Page Design Consideration and Principles, Search and Meta Search Engines, WWW, Browser, HTTP, Publishing Tools	1
7.	**HTML Programming Basics** HTML page structure, HTML Text, HTML links, HTML document tables, HTML Frames, HTML Images, multimedia, Interactivity Tools 08 Hrs. 33 ASP, VB Script, JAVA Script, JAVA and Front Page, Flash	1
8.	**Internet Security Management Concepts** Information Privacy and Copyright Issues,Overview of Internet Security, Firewalls, Internet Security, Management Concepts and Information Privacy and Copyright Issues, basics of asymmetric cryptosystems.	1

Text Books

1. Greenlaw R and Hepp E "Fundamentals of Internet and www" 2nd EL, Tata McGrawHill,2007.

2. . Ivan Bayross, "HTML, DHTML, JavaScript, Perl CGI", 3rd Edition, BPB Publications.

3. D. Comer, "The Internet Book", Pearson Education, 2009.

References

1. M. L. Young,"The Complete reference to Internet", Tata McGraw Hill, 2007.

2. Godbole AS & Kahate A, "Web Technologies", Tata McGrawHill,2008.

3. Jackson, "Web Technologies", Pearson Education, 2008. 4. B. Patel & Lal B. Barik, " Internet & Web Technology ", Acme Learning

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	RDBMS	**Semester**	MTech	
Department	Computer Science & Engineering	**Course Code**	CST502	
Credits	03	**L**	**T**	**P**
Course Type	Theory	2	0	2

Course Objectives

1. Understand functional components of the DBMS.
2. Devise queries using Relational Algebra, Relational Calculus and SQL.
3. Develop E-R model and design database schema
4. Understand transaction processing, concurrency control and recovery techniques.

Learning Outcomes

The student should develop skills and understanding in:

- the design methodology for databases and verifying their structural correctness
- implementing databases and applications software primarily in the relational model
- applying the theory behind various database models and query languages
- implementing security and integrity policies relating to databases

Course Synopsis

The **course** emphasizes the understanding of the fundamentals of relational systems including data models, database architectures, and database manipulations.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction to databases**: What is database system, purpose of database system, view of data, relational databases, database architecture, transaction management,	1
2.	**Data models**: The importance of data models, Basic building blocks, Business rules, The evolution of data models, Degrees of data abstraction.	1
3.	**Database design and ER Model**: overview, ER-Model, Constraints, ER-Diagrams, ERD Issues, weak entity sets, Codd's rules, Relational Schemas,	1
4.	**Relational database model**: Logical view of data, keys, integrity rules. Relational Database design: features of good relational database design, atomic domain and Normalization (1NF, 2NF, 3NF, BCNF).	2
5	**Relational Algebra and calculus**: Relational algebra: introduction, Selection and projection, set operations, renaming, Joins, Division, syntax, semantics. Operators, grouping and ungrouping, relational comparison. Calculus: Tuple relational calculus, Domain relational Calculus, calculus vs algebra, computational capabilities.	2

6	**Constraints, Views and SQL**: What is constraints, types of constrains, Integrity constraints, Views: Introduction to views, data independence, security, updates on views, comparison between tables and views SQL: data definition, aggregate function, Null Values, nested sub queries, Joined relations. Triggers.	1
7	**Transaction management and Concurrency control**: Transaction management: ACID properties, serializability and concurrency control, Lock based concurrency control (2PL, Deadlocks), Time stamping methods, optimistic methods, database recovery management.	2
Text Books		
1.	Elamsri, Navathe, Somayajulu and Gupta, Fundamentals of Database Systems, 6th Edition, Pearson Education, 2011.	
2.	Rob, Coronel, "Database Systems", Seventh Edition, Cengage Learning.	
References		
2.	A Silberschatz, H Korth, S Sudarshan, "Database System and Concepts", fifth Edition McGraw-Hill ,	

National Institute of Technology Srinagar

8th Semester

Computer Science & Engineering

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Project	**Semester**	8^{th}	
Department	Computer Science & Engineering	**Course Code**	CSP460	
Credits	12	**L**	**T**	**P**
Course Type	Practical	0	0	10

Course Objectives

The aim of the Final Year Project is to enhance the student's knowledge and skills in solving problem through structured project research in order to produce a competent and productive engineer.

Learning Outcomes

Upon completion of Final Year Project, student should be able to:

- Identify and describe the problem and scope of project clearly.
- Collect, analyze and present data into meaningful information using relevant tools.
- Select, plan and execute a proper methodology in problem solving.
- Work independently and ethically.
- Present the results in written and oral format effectively
- Identify basic entrepreneurship skills in project management

Course Synopsis

Final Year Project (FYP) is the individual project, which takes place over the two semesters. It is a mandatory subject for students to be awarded with Bachelor Degree. In the FYP, students are expected to undergo research studies which relate to the major course offered in the faculty.

The final year project consists of two parts; the first part FYP 1 is a prerequisite to the second part, FYP 2.

These two parts of FYP are under subject code CSE-705 and CSE-801 respectively.

Project Report Structure

The project group should spend enough time to prepare the report. It deals with all major issues that should be taken in to account in the report. In a report, not only there is a need of having proper structures but also it should speak about the format of the text processing. It has several sections. In the first section, it deals with the some initial pages such as Title Page, Declaration, Certificate, Acknowledgement, and Table of Contents.

In Section 2 the way one can write Abstract of the report has been explained.

Section 3 discusses the expectation of a reader in the first chapter of the report known as Introduction.

Next section is used to discuss the expected contents of the middle chapters of the report. Generally there are at least three middle chapters in the report such as Methodology, Results and Discussion.

Last chapter of the report discusses Conclusion and Future Work. Its format and expectation is given in Section 5.

Next section deal with the situation when one has to add Appendices in the project report. Last section explains the way one should write the references that are used in writing the report.

1. Preliminary Pages

This section discusses the need of having preliminary pages along with their need. Preliminary pages are Title Page, Certificate, Declaration, Acknowledgement, Table of Contents.

a) In **Title Page** author must mention Title of the report along with name of project group, supervisor, Department, Month and Year of submission. An effort should be made so that the Title of the report consists of minimum number words but also it can give enough meaning on the work studied in the report. The desired format of the Title Page is enclosed in Appendix. The project group is advised to check it carefully.

b) The draft copy of the project report has to be submitted to the supervisor for the review. Based on the comments given by him, the project group should modify the report and include a **certificate** signed by the supervisor. The certificate should mention that the work has been carried out by this project group and has not been submitted by any other group of the institute for the award of any other degree. The format of the certificate is given in the Specimen Format of the report.

c) The project report must also contain a **declaration** from the project group to avoid the problem of plagiarism. Format of the certificate is given in the Specimen Format of the report. It is expected that the project group has submitted the results of their own thought, research, or self-expression. In cases where students feel unsure about a question of plagiarism involving their work, they are obliged to consult their instructors on the matter before submission.
When students submit work purporting to be their own, but which in any way borrows ideas, organization, wording or anything else from another source without appropriate acknowledgment of the fact, the students are guilty of plagiarism.
Plagiarism includes reproducing someone else's work, whether it be a published article, chapter of a book, a paper from a friend or some file, or whatever. Plagiarism also includes the practice of employing or allowing another person to alter or revise the work which a student submits as his/her own, whoever that other person may be. Students may discuss assignments among themselves or with an instructor or tutor, but when the actual work is done, it must be done by the student, and the student alone.
When a student's assignment involves research in outside sources or information, the student must carefully acknowledge exactly what, where and how he/she has employed them. If the words of someone else are used, the student must put quotation marks around the passage in question and add an appropriate indication of its origin. Making simple changes while leaving the organization, content and phraseology intact is plagiaristic. However, nothing in these Rules shall apply to those ideas which are so generally and freely circulated as to be a part of the public domain.
Plagiarism is a serious academic offense. Students occasionally plagiarize unknowingly, but this ignorance of the definition of plagiarism is not an excuse and does not prevent a penalty from being applied.

d) Most thesis authors put in a page of thanks, known as **Acknowledgements**, to those who have helped them in matters scientific, and also indirectly by providing such essentials as food, education, genes, money, help, advice, friendship etc. If any work is collaborative, one should make it quite clear who did which sections. Author should not forget to acknowledge if he has obtained any financial support from the Department or any other source to pursue this work. Format is enclosed.

e) Table of Contents provides the list of all headings and subheadings with page numbers. Subheadings are properly indented. It also provides list of all Tables, Figures and Symbols used in the report along with their respective page numbers. A general format is given in the Specimen Format of the report.

2. Abstract

An **abstract** is an abbreviated version of the project report. It should be limited to a maximum of 250 words. The project abstract appears at the beginning of the report. Almost all scientists and engineers agree that an abstract should have the following five pieces such as Introduction, Problem Statement, Procedure, Results and Conclusion. In **Introduction**, one describes the purpose for doing such a project. It should address the need for such type of work. It should explain something that should cause people to change the way they go about their daily business. If the project leads to an invention or development of a new procedure, it should mention its advantages. Abstract should be such that it motivates the reader to read the entire paper or display board. In the next stage, one should write down the **Problem Statement**. It is needed to identify the problem that has been considered in the project. In **Procedures, t**he approach used to investigate the problem should be mentioned in the abstract. However it is expected that one will not go into detail about materials unless they were critical to the success. However one should try to describe the most important variables used in the report if room permits. In the fourth stage, abstract must clearly state the **Results**/ achievements obtained through the execution of the project. Finally **Conclusions** are given an the last stage. One should state clearly whether the objectives have been met or not. If not, the reasons behind it should be stated in few words.

In writing the abstract, one should avoid the following things:

- Any technical term that most readers do not understand should be avoided.
- Any abbreviation or acronym that is not commonly used should be avoided.
- Abstracts should not contain a bibliography or citations.
- Abstracts should not contain any tables or graphs.
- Abstracts must give only minimal reference to any earlier work.
- Abstract should only include procedures done by the group.
- One should not put acknowledgements in the abstract.

2.2.1 Importance of Abstract

Abstract helps people quickly to determine if they want to read the entire report. Consequently, at least ten times as many people will read an abstract as any other part of the total work. It's like an advertisement of the work done. If one wants judges and the public to be excited about the project, then he must write an exciting, engaging abstract.

Since an abstract is so short, each section is usually only one or two sentences long. Consequently, every word is important to conveying the message. If a word is boring or vague, one should refer to a thesaurus and find a better one. If a word is not adding something important, one should not use it. But, even with the abstract's brief length, one should not be afraid to reinforce a key point by stating it in more than one way or referring to it in more than one section.

2.2.2 Meeting the Word Limit

Most authors agree that it is harder to write a short description of something than a long one. One can use the following tip. For the first draft, one should not be overly concerned about the length. Just he should make sure that the draft abstract has considered all the key information. Then one can cross out words, phrases, and sentences that are less important than others. In the next stage one can look for places where sentences can be combined so that the total length can be reduced. After a short break, one should again read the draft to give the final shape of the abstract. With a fresh eye, one will probably find new places to cut and will be able to make the abstract.

3. Introduction

It is the first chapter of the Report. The purpose of an introduction in the B. Tech Project Report is to justify the reasons for writing about the report. The goal in this section is to introduce the topic to the reader, provide an overview of previous research on the topic, and identify the own hypothesis. The goals just mentioned could, if abused, lead to write an introduction that is pages and pages long. It can be noted here that the introduction should not contain every bit of detail in the report, and it should not include support for the report. An introduction might, however, include the reasons for supporting the report.

In order for readers to trust the writer, the introduction must be well written with few errors. In order to keep readers reading, the writer needs to catch the attention of the reader and write in an interesting way. The unique language enhancement feature may suggest words to strengthen the writing. Strong writing may hold readers' attention.

In addition to well-written English with strong vocabulary, there are a few other strategies to hold readers' attention. It should be noted that to excite the readers' interest, one may also want to sound as though the readers know the topic that are considered in the report. Some of the following strategies in the bullet-list above may help.

- To pose a specific question that can invite the readers to keep reading for the answer--A provocative question works well to engage readers, so long as it doesn't put them off.
- To choose statistics to surprise readers or to go against the common belief about a topic.
- To mention a short, interesting anecdote (or story) related to the topic.
- To provide an interesting (and relevant) quote.
- To develop an unusual or unexpected comparison.

It has been observed that the difficulty level to write the Introduction is average and it takes the variable amount of time. Generally, it consists of the five major tasks and they are

- Problem introduction
- Previous work carried out in this field.
- Researching the problem studied.
- Summarization of the results obtained.
- Organization of the report.

3.1 Introduction of the Problem

This is the first task which provides a brief description of the research question. It speaks about the type of the experiment or study attempting to demonstrate or the phenomena that are planned to study. It also should provide a brief history of the problem and explain how it is important to make such type of study.

3.2 Summarize Previous Research

The second task of the Introduction is to provide a well-rounded summary of previous research that is relevant to the problem considered for the study. So, before one begins to write this summary, it is important to thoroughly research the problem. Finding appropriate sources amid thousands of journal articles can be a daunting task, but there are a number of steps that one can take to simplify the work which has been carried out.

3.3 Researching the Problem

One should search a journal database to find articles on the considered subject. Once an

article is found, it is suggested to look at the reference section to locate other studies cited in the article. While taking notes from these articles, one should be sure to write down all the desired information. A simple note detailing the author's name, journal, and date of publication can help to keep track of sources and to avoid plagiarism.

3.4 Provide the Hypothesis

Once the previous research work has been summarized, it is expected that a subsection is to be written to explain areas where the research work is lacking or potentially flawed. It should also mention the missing or unrevealed components from previous studies on the considered problem. As a result, the derived hypothesis should lead from these questions. At the end of the Introduction, one should clearly state the hypothesis and describe the expectation that one wants to achieve through the experiment or study.

3.5 Organization of the Report

This is the last task in this section. One should write clearly the organization of the project. It provides the short description of the work reported in each chapter.

3.6 Other Issues

Types of introductions that should be avoided are given below.

3.6.1 The Dictionary Definition

Many papers begin with defining words through the dictionary meaning and then continue to discuss the topic. This type of introduction has become very stale with faculty, who have seen it thousands of times.

3.6.2 Cutting to the Chase too Quickly

It is too easy to go too far while avoiding overly general introductions. Avoid jumping right into a thesis statement and do not try to cover every topic in the first paragraph. It is difficult to say how specific to be in an introduction, but consider the idea that this part of a paper provides "the lay of the land" for a reader who can then know *why* the paper is worth finishing.

3.6.3 Memorable Quotations

Some readers do not like papers to start with another's words. This overused strategy may be acceptable if a direct quotation sets the stage for what follows and its relevance is discussed in the introduction.

3.6.4 The "telegraphic" sentence

Here a writer uses the first or second or third person to tell a reader what is going to happen. This should be avoided as far as possible. One can use passive voice to avoid such type of situation.

3.6.5 Use of Tense

Very often, it has been seen that author mixes up various tenses to express. It is expected that one uses only present tense through the report.

3.7 Purpose of Introduction

Thus an introduction serves two purposes:

- It gives readers an idea of what the rest of the writing will say.
- It provides a reason for readers to keep reading.

The first reason is the most important. Some methods for letting readers know include:

- A clear and specific thesis statement (see How to Write a Thesis Statement).
- Providing background or history for the topic.
- Defining a term, phrase, or concept central to the writing.
- Providing statistics

Here is a writing tip for Introduction. One should not write Introduction until draft of the report has been revised and finalized. That way, one can have a better idea of what the work actually says, instead of what one can think what it will say.

4. Middle Chapters

In some theses, the middle chapters are the articles of which the student group is the major author. There are several disadvantages to this format. One is that a report is both allowed and expected to have more detail than a journal article. For journal articles, one usually has to reduce the number of figures. In many cases, all of the interesting and relevant data can go in the project report, and not just those which appeared in the journal. The degree of experimental detail is usually greater in a project report. Relatively often a researcher requests a report in order to obtain more detail about how a study was performed.
Another disadvantage is that the journal articles may have some common material in the introduction and the "Materials and Methods" sections.

4.1 Structure of Report

The exact structure in the middle chapters may vary among theses. In some reports, it is necessary to establish some theory, to describe the experimental techniques, then to report what has been done on several different problems or different stages of the problem, and then finally to present a model or a new theory based on the new work. For such a report, the chapter headings might be: Theory, Materials and Methods, {first problem}, {second problem}, {third problem}, {proposed theory/model} and then the conclusion chapter. For other reports, it might be appropriate to discuss different techniques in different chapters, rather than to have a single Materials and Methods chapter.
Following are the some comments on the elements Materials and Methods, Theory, Results and Discussion which may or may not correspond to thesis chapters.

4.2 Materials and Methods

This varies enormously from project report to report, and may be absent in theoretical reports. It should be possible for a competent researcher to reproduce exactly what author has done by following the description. There is a good chance that this test can be applied: sometime after the author has left the institution, another researcher may like to do a similar type of experiment either with author's help, or on a new set-up in a different institute. Author should write clearly this chapter for the benefit of that researcher.
In some reports, particularly multi-disciplinary or developmental ones, there may be more than one such chapter. In this case, the different disciplines should be indicated in the chapter titles.

4.3 Theory

When author is reporting theoretical work that is not original, he should include sufficient material to allow the reader to understand the arguments used and their physical bases. Sometimes he may be able to present the theory *ab initio*, but he should not reproduce two

pages of algebra that the reader could find in a standard text. One should not include theory which is not related to the work done.

When writing this section, author should concentrate at least as much on the physical arguments as on the equations. He should try to explain the meaning of each equation and provide sufficient explanation on the important ones.

When author is reporting his own theoretical work, he must include rather more detail, but he should consider moving lengthy derivations to appendices. He should think too about the order and style of presentation: the order in which he has done the work may not be the clearest presentation.

Suspense is not necessary in reporting science. Author should tell the reader where he is going before he starts. The following are the expectation from this chapter:

- Information to allow the reader to assess the believability of the results.
- Information needed by another researcher to replicate the experiment.
- Description of the materials, procedure, theory.
- Calculations, technique, procedure, equipment, and calibration plots.
- Limitations, assumptions, and range of validity.

It should also carefully see the following issues:

- One should be able to replicate the study accurately (for example, all of the optional and adjustable parameters on any sensors or instruments that were used to acquire the data).
- Another researcher should be able to find accurately and reoccupy the sampling stations or track lines.
- There should be enough information provided about any instruments used so that a functionally equivalent instrument could be used to repeat the experiment.
- If the data is in the public domain, another researcher should be able to lay his or her hands on the identical data set.
- One should be able to replicate any laboratory analyses that have been used.
- One should be able to replicate any statistical analyses.
- Another researcher should be able to approximately replicate the key algorithms of any computer software.

Citations in this section should be limited to data sources and references of where to find more complete descriptions of procedures. Results are not to be discussed in this chapter.

4.4 Results and Discussion

The results and discussion are very often combined in reports. This is sensible because of the length of a report: one may have several chapters of results and, if one waits till they are all presented before he begins discussion, the reader may have difficulty remembering what he is talking about. The division of Results and Discussion material into chapters is usually best done according to subject matter.

One must ensure that he has described the conditions which obtained for each set of results. The report must explain clearly the way of obtaining the constant, other relevant parameters. He should be sure too that he has used appropriate statistical analyses. Where applicable, he Has shown measurement errors and standard errors on the graphs. It is expected that appropriate statistical tests have been used in the work.

4.4.1 Graph Plotting

Author should take sufficient care to plot graphs. The origin and intercepts are often

important so, unless the ranges of data make it impractical, the zeros of one or both scales should usually appear on the graph. One should show error bars on the data, unless the errors are very small. For single measurements, the bars should be the best estimate of the experimental errors in each coordinate. For multiple measurements these should include the standard error in the data. The errors in different data are often different, so, where this is the case, regressions and fits should be weighted (i.e. they should minimize the sum of squares of the differences weighted inversely as the size of the errors.) A common failing in many simple software packages that draw graphs and do regressions is that they do not treat errors adequately. Author can just 'paste' data into the input and it generates a .ps file of the graph.
In most cases, each result needs discussion. Meaning of each result has to be explained. It should be explained clearly how they fit into the existing body of knowledge. Consistency of results is to be seen. Results should give new insights. If possible, it should suggest new theories or mechanisms. Some salient points which should be taken care are given below.

- The results are actual statements of observations, including statistics, tables and graphs.
- Actual information on range of variation.
- Discussion of both negative results as well as positive. Not to interpret results at this stage.
- Availability of sufficient details of results so that others can draw their own inferences and construct their own explanations.
- Use of S.I. units (m, s, kg, W, etc.) throughout the report.
- Breaking up of results into logical segments by using subheads

The author must make it crystal clear to the reader which statements are observation and which are interpretation. In order to achieve this, it is suggested to have two different chapters- one for results and other for discussion. Overlay interpretation on top of data in Figures should be avoided.

4.5 Discussion

Author can start with a few sentences that summarize the most important results. The discussion section should be a brief essay in itself, addressing the following issues:

- The major patterns in the observations that can be referred as spatial and temporal variations.
- The relationships, trends and generalizations among the results.
- Expectations to these patterns or generalizations, if any.
- Likely causes (mechanisms) underlying these patterns resulting predictions.
- Agreement or disagreement with previous work.
- Interpretation of results in terms of background laid out in the introduction - the relationship of the present results to the original problem.
- Implication of the present results for other unanswered problems.
- Multiple hypotheses: There are usually several possible explanations for results. One should be careful to consider all of these rather than simply pushing the favorite one. If one can eliminate all but one, that is great, but often that is not possible with the data in hand. In that case one should give even treatment to the remaining possibilities, and try to indicate ways in which future work may lead to the discrimination.
- Avoiding bandwagons: A special case of the above. It is suggested to avoid jumping a currently fashionable point of view unless the results really do strongly support them.
- Things that are known or understood after reading the report.

- Inclusion of the evidence or line of reasoning supporting each interpretation.
- Significance of the present results.

This section should be rich in references to similar work and background needed to interpret results. However, interpretation/discussion section(s) are often too long and verbose. There may be some material that does not contribute to one of the elements listed above. In that case, this may be material that one may like to consider deleting or moving. It is suggested to break up the chapter into logical segments by using subheads.

4.6 Conclusions and Further Work

This is the last chapter of the project report. Abstract of the report should include conclusions in very brief form, because it must also include some other material. A summary of conclusions is usually longer than the final section of the abstract, and author has the space to be more explicit and more careful with qualifications. He might find it helpful to put the conclusions in point form.
It is often the case with scientific investigations that more questions than answers are produced. It must indicate whether the work carried out suggests any interesting further avenues. It should discuss the possibility of improving the work by future workers. A paragraph should be written on the practical implications of the work.
This chapter should usually be reasonably short---a few pages perhaps. As with the introduction, it is a good idea to ask someone who is not a specialist to read this section and to comment.

4.7 Appendices

If there is material that should be in the project report but which would break up the flow or bore the reader unbearably, include it as an appendix. Some things which are typically included in appendices are: important and original computer programs, data files that are too large to be represented simply in the results chapters, pictures or diagrams of results which are not important enough to keep in the main text. Thus in the appendix, one should include

- all data used in the report
- reference data/materials not easily available
- tables (where more than 1-2 pages)
- calculations (where more than 1-2 pages)
- all key articles
- list of all additional resource materials
- list of equipment used for an experiment or details of complicated procedures.

4.8 References

It is tempting to omit the titles of the articles cited, and the university allows this, but thinks of all the times when author has seen a reference in a paper and gone to look it up only to find that it was not helpful after all. If he cites a journal article or book, the reader can go to a library and check that the cited document and check whether or not it says what he says it did. A web site may disappear, and it may have been updated or changed completely. So references to the web are usually less satisfactory. Nevertheless, there are some very useful and authoritative sources. However it is expected that such citations should not be overused. In particular, a web citation should not be used if there exists a "hard" citation. Author should

give the exact URL. Thumb rules followed to refer some one's work are given below.

- cite all ideas, concepts, text, data that are not own by the project group
- if author makes a statement, he must back it up with his own data or a reference
- all references cited in the text must be listed
- list all references cited in the text in alphabetical

National Institute of Technology
Srinagar

Detailed Syllabus

Electives

Computer Science & Engineering

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Simulation and Modelling	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST001	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

The objectives of this course are to:

- Introduce students to the simulation and modelling techniques
- Provide students with opportunities to develop basic simulation and modelling skills with respect to carrying out research projects using any simulation method on the computer.

Learning Outcomes

After the course the student should be able to

- Define basic concepts in modelling and simulation
- Classify various simulation models and give practical examples for each category
- Construct a model for a given set of data and motivate its validity
- Analyze output data produced by a model and test validity of the model

Course Synopsis

In this course students study the representation and simulation of physical systems using a range of mathematical formulations. Case studies are used to illustrate a variety of modelling techniques. The students learn to develop typical mathematical models and utilise them to predict the behaviour of common industrial and engineering systems.

Course Outline / Content

Unit	Topics	Week
1.	Mathematical Model, types of Mathematical models and properties	1
2.	Procedure of modeling, Graphical method: Barterning model	1
3.	Basic optimization, Basic probability: Monte-Carlo simulation	1
4.	Approaches to differential equation: Heun method	1
5.	Local stability theory: Bernoulli Trials, Classical and continuous models	1
6.	Case studies in problems of engineering and biological sciences	1
7.	General techniques for simulating continuous random variables	2
8.	simulation from Normal and Gamma distributions	2
9.	simulation from discrete probability distributions	2
10.	simulating a non – homogeneous Poisson Process and queuing system	2

Text Books

1.	Frank R. Giordano, William P. Fox,First Course in Mathematical Modeling, Cengage Learning
2.	1. A.M.Law and W.D.Kelton, Simulation Modeling and Analysis, T.M.H. Edition.
References	
1.	A. C. Fowler, Mathematical Models in Applied Sciences, Cambridge University Press.
2.	S.M. Ross, Simulation, India Elsevier Publication

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	**Graph Theory**	**Semester**		
Department	CSE	**Course Code**	CST002	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- To learn the basic terminology and results concerning graphs.
- To learn proof techniques and algorithms involving graphs.
- To learn how to apply computer programs to study graphs.
- To provide an acquaintance with mathematical notation used to express physical and Natural laws.
- To learn about open problems in graph theory.

Learning Outcomes

Upon completion of this course, students will be able to do the following:

- Have a strong background of graph theory which has diverse applications in the areas of computer science.
- Solve problems using basic graph theory.
- Determine whether graphs are Hamiltonian and/or Eulerian.
- Solve problems involving vertex and edge connectivity, planarity and crossing numbers.
- Model real world problems using graph theory.

Course Synopsis

Graph Terminology; Types of graphs; Trees and its types; Spanning Trees ; Algorithms for MST; Planar and Dual Graphs; Matrix representation of Graphs; Coloring, Covering and partitioning; Graph theoretic Algorithms.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction:** Graph Terminology, Incidence and Degree**,** Isolated vertex, pendant vertex and Null Graph, Isomorphism, Walks, Paths and Circuits, Connected Graphs, Disconnected graphs and Components, Euler Graphs, Operations on graphs, Hamiltonian paths and circuits, The Travelling salesman problem, Konigsberg bridge problem**,** Three utility problem.	2
2.	**Trees:** Properties of Trees**,** Distance and Centres in a tree**,** Rooted and Binary Trees**,** Spanning Trees. Algorithms for finding minimal spanning tree: Kruskal's algorithm, Prim's algorithm. **Cut-sets and Cut-Vertices:** Cut-sets, All cut-sets in a graph, Fundamental Circuits and cut-sets, connectivity and separability, Network Flows, 1-isomorphism, 2-isomorphism.	3
3.	**Planar and Dual Graphs:** Planar Graphs, Kuratowski's two graphs, Kuratowski's Theorem, Detection of planarity, Geometric dual, Combinatorial dual.	3

	Matrix Representation of Graphs: Incidence matrix, Circuit matrix, Cut-set matrix, path matrix and Adjacency matrix.	
4.	**Coloring, Covering, and Partitioning:** Chromatic number, Chromatic partitioning, Chromatic polynomial, Matching, Coverings, The Four Color problem.	2
5.	**Directed Graphs:** Types of digraphs, Euler Digraphs, Trees with directed edges, Matrix representation of digraphs, Tournaments, Acyclic digraphs and decyclization.	2
6.	**Graph theoretic Algorithms:** Shortest path algorithms, Dijkstra's algorithm, Warshall - Floyd algorithm, Depth-First search in a graph, Breadth – first search in a graph.	2
Text Books		
1.	NarsinghDeo, Graph Theory with Applications to Engineering and Computer Science, PHI.	
2.	R.J. Wilson,Introduction to Graph Theory, Fourth Edition, Pearson Education, 2003	
References		
1.	Douglas B. West, "Introduction to Graph Theory", Prentice Hall of India, 2005	
2.	S.Even. Graph Algorithms, Computer Science Press,1979	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Digital Signal Processing	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST003	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course includes:

- To develop methods for processing discrete-time signals.
- To understand the processes of A-D and D-A conversion.
- To acquire some familiarity with digital filters in terms design and implementation and to become familiar with how various types of filters affect signal characteristics.
- To understand the discrete Fourier transform and discrete spectral analysis.
- To become familiar with some applications of digital processing.

Learning Outcomes

After completion of course students will be able to:

- Be able to perform FIR AND IIR filters by hand to meet specific magnitude and phase requirements.
- Perform Fourier transform and inverse Fourier transform using definitions, tables of standard transforms and properties.
- Design and implement digital filters by hand and by using Matlab.
- Use computers and Matlab to create, analyse and process signal and to simulate and analyse systems sound and image synthesis and analysis.

Course Synopsis

Digital Signal Processing discusses analysis and representation of discrete-time signal systems, including discrete-time convolution, difference equations, the z-transform, and the discrete-time Fourier transform. The course proceeds to cover digital network and non recursive (finite impulse response) digital filters. Digital Signal Processing focuses on digital filter design and a discussion of the fast Fourier transform algorithm for computation of the discrete Fourier transform.

Course Outline / Content

Unit	Topics	Week
1.	**Discrete Time Signals And Systems :**Representation of discrete time signal – classifications – Discrete time – system – Basic operations on sequence – linear – Time invariant – causal – stable – solution to difference equation – convolution sum – correlation – Discrete time Fourier series – Discrete time Fourier transform.	2
2.	**Fourier And Structure Realization:** Discrete Fourier transform	

	– properties – Fast Fourier transform – Z-transform – structure realization – Direct form – lattice structure for FIR filter – Lattice structure for IIR Filter.	3
3.	**Filters:** FIR Filter – windowing technique – optimum equiripple linear phase FIR filter – IIR filter – Bilinear transformation technique – impulse invariance method – Butterworth filter – Tchebycheff filter.	3
4.	**Multistage Representation:** Sampling of band pass signal – anti aliasing filter – Decimation by an integer factor – interpolation by an integer factor – sampling rate conversion – implementation of digital filter banks – sub-band coding – Quadrature mirror filter – A/D conversion – Quantization – coding – D/A conversion – Introduction to wavelets.	3
5.	**Digital Signal Processors:** Fundamentals of fixed point DSP architecture – Fixed point number representation and computation – Fundamentals of floating point DSP architecture – floating point number representation and computation – study of TMS 320 C 54XX processor – Basic programming – addition – subtraction – multiplication – convolution – correlation – study of TMS 320 F2XXX processor – Basic programming – convolution – correlation.	3
Text Books		
1.	John G. Proakis, Dimitris, G. Manolakis, "Digital Signal Processing: Principles, Algorithms and Applications", PHI.	
2.	S.Salivahanan, A.Vallavaraj and C.Gnanapriya, "Digital Signal Processing", TMH, 2000.	
3.	A.V. Oppenheim and R.W.Schafer, Englewood, "Digital Signal Processing", Prentice-Hall Inc, 1975. 4.	
1.	B.Venkatramani&M.Bhaskar, "Digital Signal Processors architecture, programming and applications",TMH, 2002.	
References		
1.	Rabiner L.R and C.B Gold,"Theory and Applications of Digital Signal Processing", Prentice Hall of India,1987.	
2.	Leudeman L.C, "Fundamentals of Digital signal processing", Harper & Row Publication, 1986.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Multimedia Technology	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST004	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- To provide the foundation knowledge of multimedia computing, e.g. media characteristics, compression standards, multimedia representation, data formats, multimedia technology development.
- To provide programming training in multimedia computing, multimedia system design and implementations.

Learning Outcomes

Upon completion of this course, the students will be able to do the following:

- Should be able to take into considerations in multimedia techniques design and implementation.
- Should have understood the characteristics of human's visual system, human are audio system.
- Should be able to design and develop multimedia systems according to the requirements of multimedia applications.
- Program multimedia data and be able to design and implement media applications.

Course Synopsis

Basics of Multimedia Systems; Architecture and its components; Data acquisition, sampling, quantization and compression of audio and speech; Image and video representation and compression standards; Fundamentals of Multimedia Communication and Networking; Hypermedia Presentation; Multimedia Information Systems.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction to Multimedia Systems:** Architecture and components, multimedia distributed processing model, synchronization, orchestration and quality of service architecture.	1
2.	**Audio and Speech:** Data acquisition, sampling and quantization, human speech production mechanism, digital model of speech Production, analysis and synthesis, psycho-acoustics, low bit rate speech compression, MPEG audio compression.	2
3.	**Images and Video:** Image acquisition and representation, composite video signal, NTSC, PAL and SECAM video standards; Bi-level image compression standards, JPEG and MPEG.	2
4.	**Multimedia Communication:** Fundamentals of data communication and networking, bandwidth requirements of different media; Real time constraints: Audio latency, video data	3

	rate; Multimedia over LAN and WAN, multimedia conferencing.	
5.	**Hypermedia Presentation:** Authoring and publishing, linear and nonlinear presentation, structuring information, different approaches of authoring hypermedia documents, hypermedia data models and standards.	3
6.	**Multimedia Information Systems:** Operating system support for continuous media applications, limitations of OS, new OS support, media stream protocol, file system support for continuous media, data models for multimedia and hypermedia information, content based Retrieval of unstructured data.	3
Text Books		
1.	Li, Z.N. and Drew, M.S., "Fundamentals of Multimedia", Pearson Education.	
2.	Hillman, D., "Multimedia Technology and Application", Galgotia Publication.	
References		
1.	Steinmetz, R., "Multimedia Computing, Communication and Applications", Pearson Education.	
2.	Buford, J., "Multimedia Systems", Addison Wesley.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Logic Programming.	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST005	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- To develop an understanding of basic knowledge and practical experience in logic programming.
- To learn about the formal concepts used as a theoretical basis for logic programming.
- To interpret problems in a style that suits logic programming
- To understand principles of declarative specification, and its relation to procedural realisations.

Learning Outcomes

Upon completion of this course, students will be able to do the following:

- List, define and apply the fundamental concepts of logic programming.
- Manually analyze and execute a given simple logic program. The analysis covers correctness and efficiency.
- Formulate simple problems in logic that can be executed as a logic program.
- Implement simple algorithms and data structures as correct logic programs.
- Take advantage of the strengths of logic programming (unification, backtracking and grammar programming) for solving simple combinatorial problems and for natural language processing.
- Explain how logic programming differs from other programming paradigms.

Course Synopsis

Formulation of problems using Proposition Logic; Rules of natural deduction; Properties of Axiomatic systems; Fundamentals of Predicate Logic; Semantic Tableaux and Resolution in Predicate Logic; Programming in Prolog; Meta level programming; Lazy and Eager Evaluation Strategies.

Course Outline / Content

Unit	Topics	Week
1.	**Proposition Logic:** Introduction of logic and functional paradigm, propositional concepts, semantic table, problem solving with semantic table.	2
2.	**Natural Deduction and Axiomatic Propositional Logic:** Rules of natural deduction, sequent calculus, axiomatic systems, Meta theorems. Important properties of AL (Axiomatic Logic), resolution, resolving arguments.	2
3.	**Introduction to Predicate Logic:** Objects, predicates and quantifiers, functions, first order language, quantifiers, scope and binding, substitution. An axiomatic system for first order predicate logic, soundness and completeness, Axiomatic semantics and programming.	3
4.	**Semantic Tableaux & Resolution in Predicate Logic:** Semantic tableaux, instantiation rules, problem-solving in predicate logic, normal forms. Herbrand universes and H-interpretation, resolution, unification, resolution as a computing tool.	2

5.	**Prolog Concepts:** Programming in Prolog (overview), Meta level programming and Meta interpreters. Nondeterministic programming, incomplete data structure, second order programming in Prolog. Logic grammars: definite clause grammar, A grammar interpreter.	2
6.	**Lazy and Eager Evaluation Strategies:** Evaluation strategies, Lazy evaluation: evaluation order and strictness of function, programming with lazy evaluation, interactive functional program, delay of unnecessary computation, infinite data structure, eager evaluation and reasoning.	3
Text Books		
1.	John Kelly, "The Essence of Logic", Prentice-Hall India.	
2.	SarojKaushik, "Logic and Prolog Programming", New Age International ltd.	
3.	TasamiHagiya and Philip waddle, "Functional and Logic Programming", 8th Edition, 2006.	
References		
1.	TestsuoIda,Atsushiohori and Masato Takichi, "Functional and Logic Programming", 2006.	
2.	Chang, C.L and Lee R.C .T, "Symbolic Logic and Mechanical theorem proving", Academic Press, New York, 2006.	
3.	J.W. Lloyed, Springer Verlog, "Foundation of logic programming", New York, 2/E, 1987.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Embedded Systems	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST006	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

The aim of this course to provide the student with a detailed understanding of to Microcontrollers and Embedded systems. The course covers fundamentals, The 8051 Architecture, Assembly Language Programming, Instruction set, Serial Communication and Interfacing techniques of 8051 Microcontroller.

Learning Outcomes

- To acquire knowledge about microcontrollers embedded processors and their applications.
- Foster ability to understand the internal architecture and interfacing of different peripheral devices with Microcontrollers.
- Foster ability to write the programs for microcontroller.
- Foster ability to understand the role of embedded systems in industry.
- Foster ability to understand the design concept of embedded systems.

Course Synopsis

Introduction to real time systems, The 8051 Architecture, Memory organization, 8051 Assembly Language Programming, Instruction set, 8051 Serial Communication, Microcontroller Interfacing, Basic concept of PIC microcontroller.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction**: Concept of Real time Systems, Challenges in Embedded System Design. Introduction to Microcontrollers and Embedded Processors, Microcontrollers survey, four bit, eight bit, sixteen bit, thirty two bit Microcontrollers, Comparing Microprocessors and Microcontrollers, Overview of the 8051 family.	2
2.	**The 8051 Architecture**: Hardware, Oscillator and clock, program counter, data pointer, registers, stack and stack pointer, special function registers.	1
3.	**Memory organization**: Program memory, data memory, Input / Output Ports, External memory counter and timer, serial data Input / output, Interrupts.	2
4.	**8051 Assembly Language Programming**: Structure of Assembly language Assembling and running an 8051 program, Addressing modes, Accessing memory using various addressing modes.	2
5.	**Instruction set**: Arithmetic operations and Programs, Logical operations and Programs, Jump and Call instructions and Programs, I /O Pot Programs, Single bit instructions and Programs, Timer and counter and Programs.	2
6.	**8051 Serial Communication**: Connection to RS-232, Serial Communication Programming, Interrupts Programming.	2

7.	**Microcontroller Interfacing**: Key Board, Displays, Pulse Measurement, D / A and A/D conversion, Stepper Motor-	2
8.	**Basic concept of PIC microcontroller**: Microcontroller Architecture, PIC16F.	1
Text Books		
1.	The 8051 Microcontrollers and Embedded Systems : Muhammed Ali Mazidi	
2.	The 8051 Microcontrollers Architecture, Programming & Applications Kenneth J. Ayala.	
References		
1.	Design with PIC Microcontroller: John Petman	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Advanced Java and Android Programming	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST007	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

Build Android apps from scratch using Android Studio and Java Programming Upload your apps to Google Play and reach Millions of Android users Make Money from your apps by displaying ads.

Learning Outcomes

Upon successful completion of this class, student will be able to:

- Use the Java programming language to build Android apps
- Use the development tools in the Android development environment
- Describe the life cycles of Activities, Applications and Fragments
- Utilize Sensors like Gyroscopes, Accelerometers and GPS to add orientation and location to their apps
- Send and receive SMS messages programmatically
- Package and prepare their apps for distribution on the Google Play Store.

Course Synopsis

Collection Interfaces; Multithreading; Networking; Java Database Connectivity (JDBC).

Course Outline / Content

Unit	Topics	Week
1.	**Collections :** Collection Interfaces, Concrete Collections, The Collections Framework **Multithreading:** Creating thread and running it, Multiple Thread acting on single object, Synchronization, Thread communication, Thread group, Thread priorities, Daemon Thread, Life Cycle of Thread.	2
2.	**Networking:** Internet Addressing, Inet , Factory Methods, Instance Methods, TCP/IP Client Sockets, URL, URL Connection, TCP/IP Server Sockets, Datagram. **Enterprise Java Bean:** Preparing a Class to be a JavaBean, Creating a JavaBean, JavaBean Properties, Types of beans, Stateful Session bean, Stateless Session bean, Entity bean.	3
3.	**Java Database Connectivity (JDBC):** Merging Data from Multiple Tables: Joining, Manipulating Databases with JDBC, Prepared Statements, Transaction Processing, Stored Procedures C. **Servlets:** Servlet Overview and Architecture, Interface Servlet and the Servlet Life Cycle, Handling HTTP get Requests, Handling HTTP post Requests, Redirecting Requests to Other Resources, Session Tracking, Cookies, Session Tracking with HttpSession.	3
4.	Introduction Smart Phone Application Development. Android Architecture, User Interface Architecture, Activities and Intents, Threads, Services, Receivers and Alerts, User Interface layouts, user interface events, UI Widgets, Notification and Toast, Menus, Dialogs, Lists, Locations and Maps.	3
5.	Hardware interface-Camera, Sensors, Telephony, Bluetooth, Near Field communication, Working with Data Storage, Using Google maps, Animation	3

	and Content Providers. Network Communication, Services, Publishing your App.	
Text Books		
1.	Core and Advanced Java, Black Book, Dreamtech Press	
2.	Java SE8 for Programmers (3rd Edition) (Deitel Developer Series) by Paul Deitel and Harvey Deitel	
3.	Head First Android Development, By Dawn Griffiths and David Griffiths , OReilly.	
References		
1.	"Advanced Java 2 Platform HOW TO PROGRAM" by H. M.Deitel, P. J. Deitel, S. E. Santry – Prentice Hall	
2.	"Beginning Java™ EE 6 Platform with GlassFish 3 From Novice to Professional" by Antonio Goncalves– Apress publication	
3.	"Android Programming for Beginners", by John Horton.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	System on Chip	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST008	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course will provide an understanding of the concepts, issues, and process of designing highly integrated SoCs following systematic hardware/software co-design & co-verification Principles using state of the art synthesis and verification tools and design flows.

Learning Outcomes

Upon completion of the course, the student shall be able to:

- Understand hardware, software, and interface synthesis with emphasis on issues in interface design.
- Describe examples of applications and systems developed using a co-design approach

Course Synopsis

The incessant drive of Moore's law has created an era where most electronic systems contain chips that integrate various (hitherto discrete) components such as microprocessor, DSPs, dedicated hardware processing engines, memories, and interfaces to I/O devices and off-chip storage. Most electronic systems today - cell phones, iPods, set-top boxes, digital TVs, automobiles - contain at least one such "System-on-chip". Designing System-on-chips is a highly complex process. Before entering the traditional VLSI implementation process (RTL, logic & physical design), design teams need to perform the challenging tasks of developing a functional specification, partitioning and mapping of functions onto hardware components and software, design of communication architecture to interconnect the components, functional/performance/power analysis and validation, and more. This course will present students with an insight into the earlier stages of the System-on-chip design process (what happens before you get down to RTL, gates, transistors, and wires). In addition to the conceptual foundations, this course will also involve significant hands-on assignments and/or a project that will give students an exposure to state-of-the-art design methodologies and platforms. This course is part of a proposed "Embedded Systems" curriculum that is currently being discussed by Purdue ECE and CS.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction:**Architecture of the present-day SoC - Design issues of SoC- Hardware-Software Codesign – Core Libraries – EDA Tools.	1
2.	**Design Methodology for Logic Cores:** SoC Design Flow – guidelines for design reuse – Design process for soft and firm cores – Design process for hard cores – System Integration.	2
3.	**Design Methodology for Memory and Analog Cores:** Embedded memories – design methodology for embedded memories – Specification of analog circuits – High speed circuits.	2
4.	**Design Validation:** Core-Level validation – Core Interface verification - SoC design validation.	1

5.	**Core and SoC Design Examples :** Microprocessor Cores – Core Integration and On-chip bus – Examples of SoC.	2
6.	**Configurable Processors: A Software View:**Processor Hardware/Software Cogeneration, The Process of Instruction Definition and Application Tuning. The Basics of Instruction Extension d. The Programmer's Model .Processor Performance Factors. Example: Tuning a Large Task, Memory-System Tuning h. Long Instruction Words.	2
7.	**Configurable Processors: A Hardware View**: Application Acceleration: A Common Problem. Introduction to Pipelines and Processors. Hardware Blocks to Processors d. Moving from Hardwired Engines to Processors. Designing the Processor Interface. Novel Roles for Processors in Hardware Replacement, Processors, Hardware Implementation, and Verification Flow	2
8.	**Advanced Topics in SOC Design:** Pipelining for Processor Performance, Inside Processor Pipeline Stalls, Optimizing Processors to Match Hardware d. Multiple Processor Debug and Trace and Issues in Memory Systems.	2
Text Books		
1.	RochitRajsuman, 'System-on-a-Chip: Design and Test', Artech House, 2000.	
2.	Steve Furber, ARM System-on-Chip Architecture, 2nd ed, Addison-Wesley Professional, 2000.	
3.	D. Black, J. Donovan, SystemC: From the Ground Up, Springer, 2004.	
References		
1.	Ricardo Reis &Jochen A.G. Jess, 'Design of System on a Chip: Devices & Components', Kluwer, 2004.	
2.	Laung-Terng Wang, Charles E. Stroud, Nur A. Touba, 'System-on-Chip Test Architectures', Morgan Kaufmann, 2007	
3.	Harris, D.M. and Harris S. L., Digital Design and Computer Architecture, Morgan Kaufmann, 2007.	
4.	Pong P. Chu, RTL Hardware Design Using VHDL: Coding for Efficiency, Portability, and Scalability, John Wiley & Sons.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Advanced Internet Technologies	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST009	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course includes:

- To provide an in-depth understanding of selected Internet protocols
- To gain more advanced modelling, analysis and programming skills.
- To provide some breadth of understanding of selected computer networking topics.

Learning Outcomes

After completion of course students will be able to:

- Create sophisticated web applications for deployment to production.
- Describe the components that make up a web based application.
- Introduce security features to web applications.

Course Synopsis

The subject provides knowledge and skills in advanced internet technologies particularly related to server-side internet programming and business-to-business systems. It covers topics relevant to advanced internet programming including Web 2.0, HTML, XHTML, CSS, Javascript, Document object modelling, .NET, C#, etc..

Course Outline / Content

Unit	Topics	Week
1.	**Introduction to the Internet:** Brief overview of Internet, Internet and routing protocols, Web Server administration, Client Sever implementation, Cyber law, Search Engine Optimization Techniques, Web Based Systems.	1
2.	**Web 2.0:** Search, content networks, user-generated content, blogging, social networking, social media, tagging, social bookmarking, rich Internet applications, web services, location-based services, Web 2.0 monetization and business models, future of the Web.	1
3.	**Mark up Languages (HTML, XHTML):** HTML, dynamic HTML, XHTML syntax, headings, linking, images, special characters and horizontal rules, lists, tables, forms, internal linking, Meta elements.	1
4.	**Cascading Style Sheets (CSS):** Separation of content and presentation, inline styles, embedded style sheets, conflicting styles, linking external style sheets, positioning elements, backgrounds, element dimensions, box model and text flow, media types, building a CSS drop-down menu, user style sheets.	2
5.	**JavaScript:** Client side scripting, control statements, functions, arrays, objects, events.	1
6.	**Document object model:** Objects and collections, Extensible Markup Language (XML) and RSS: Advantages and applications, structuring data, XML namespaces, Document Type Definitions (DTDs), XML vocabularies, RSS. Other	2

	advanced internet technologies: including HTML5, JSON and JQuery.	
7.	**Introduction to .NET:** Overview of the .NET Framework - Common Language Runtime – Framework Class Library - Understanding the C# Compiler.	1
8.	**Basics of C#:** Working with Variables - Making Decisions. Classes and Objects: Methods – Properties - Interface- Partial class- Null and Casting Handling Exceptions.	1
9.	**Windows and Dialogs:** MDI – Dialogs, Lists: List Box - Tree view control - Menus and Toolbars – Delegates and Events Generics.	1
10.	**Data Access With .Net:** ADO.NET overview - Commands - Data Reader - XML Schemas - Populating a dataset. .Net Programming with SQL Server: Reading and writing streamed Xml - converting ADO.Net to Xml data.	1
11.	ASP.NET Web Forms and Controls: Web Forms Controls -Data Binding and Data Source Controls – Validation, Controls-Master and Content pages. The Asp.Net Application Environment: Configuration Files - ASP.NET, Application Security -Caching.	1
12.	**Website Creation**: Creation and hosting of websites including data connectivity.	1
Text Books		
1.	Deitel H.M. and P. J. Deitel, Internet & World Wide Web. How to Program, 4/e, Prentice Hall, ISBN 0131752421, 2008.	
2.	J. Miller, V. Kirst and Marty Stepp, Web Programming Step by Step, Step by Step Publishing; 2nd edition (2012).	
3.	Stephen C. Perry, Core C# and .NET, Prentice Hall, New Jersey.	
4.	Peter Wright, Beginning Visual C# 2005 Express Edition: From Novice to Professional, Apress.	
References		
1.	http://www2.sta.uwi.edu/~anikov/comp3400/links.htm	
2.	http://www.cs.utsa.edu/~cs4413	
3.	http://www2.sta.uwi.edu/~anikov/comp3500/lectures.htm	
4.	Mastering Computer Networks: An Internet Lab Manual", J. Liebeherr, M. El Zarki, Addison-Wesley, 2003.	
5.	A.Rodriguez, J.Gatrell, J.Karas, R.Peschkem, TCP/IP Tutorial and Technical Overview, IBM Redbook (available over the Net)	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Wireless Communication	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST010	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

An in-depth understanding of the wireless channel and the related impairments (multipath, fading), small-scale and large-scale propagation effects ,Understanding of the design of cellular systems, Detailed discussion of Multiple Access (TDMA/CDMA/OFDM), Antenna diversity, MIMO, Wireless Channel Capacity, Exposure to current and emerging wireless and cellular systems

Learning Outcomes

Learning outcomes of the course are:

- Describe and differentiate four generations of wireless standard for cellular networks.
- Determine the type and appropriate model of wireless fading channel based on the system parameters and the property of the wireless medium.
- Design wireless communication systems with key 3G and 4G technologies.

Course Synopsis

The course enables a student to understand various concepts related to antennas, radio propagation, cellular and communication and ad-hoc networks

Course Outline / Content

Unit	Topics	Week
1.	Overview of Cellular Systems and evolution 2g/3G/4G/5G	1
2.	Wireless Propagation effects and Channel Models	1
3.	Multipath fading, Shadowing, Fading margin, Shadowing margin	2
4.	Cellular Concepts – Frequency reuse, Cochannel and Adjacent channel Interference,	2
5.	C/I, Handoff, Blocking, Erlang Capacity	1
6.	Wireless propagation Part 1 - Link budget, Free-space path loss, Noise figure of receiver	2
7.	Wireless propagation	1
8.	Antenna Diversity	1
9.	Wireless Channel Capacity	1
10.	CDMA , MIMO, OFDM	2

Text Books

1.	T. S. Rappaport, “Wireless Communications – Principles and Practice” (2nd edition) Pearson, 2010
2.	Goldsmith, "Wireless Communications," Cambridge Univ Press, 2005

References

1.	J. G. Proakis, "Digital Communications," McGraw Hill
2.	Haykin&Moher, "Modern Wireless Communications" Pearson 2011
3.	A. Molisch, "Wireless Communications," Wiley, 2005

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Fault Tolerant Computing	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST011	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- To examine the concepts and techniques for redundant designs, which can make a system fault tolerant.
- To discuss the importance of fault tolerance in the design of safety critical systems.
- To examine testing techniques and algorithms in hardware, software and communications.

Learning Outcomes

After completion of this course the students should be able to:

- Explain the fundamentals and design process of fault tolerant systems,
- Understand the issues of reliability and its evaluation in the design of computer systems.

Course Synopsis

Basic concepts; Fault-Tolerant Design Techniques; reliability and availability models; Architecture of fault tolerant computers; Software fault tolerance; fault tolerance in wireless/mobile networks and Internet.

Course Outline / Content

Unit	Topics	Week
1.	**Fundamental Concepts:** Definitions of fault tolerance, fault classification, fault tolerant attributes and system structure.	2
2.	**Fault-Tolerant Design Techniques:** Information redundancy, hardware redundancy, and time redundancy.	2
3.	**Dependability Evaluation Techniques:** Reliability and availability models: (Combinatorial techniques, Fault-Tree models, Markov models), Performability Models.	3
4.	**Architecture of Fault-Tolerant Computers (case study):** General-purpose systems, high-availability systems, long-life systems, critical systems.	2
5.	**Software Fault Tolerance:** Software faults and their manifestation, design techniques, reliability models. Fault Tolerant Parallel/Distributed Architectures: Shared bus and shared memory architectures, fault tolerant networks.	3
6.	**Recent topics in fault tolerant systems:** Security, fault tolerance in wireless/mobile networks and Internet.	2

Text Books

1.	Fault Tolerant Systems by I. Koren and C.M. Krishna

References

1.	Design and Analysis of Fault-Tolerant Digital Systems Barry W. Johnson

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Image Processing	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST012	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

To learn and understand the fundamentals of digital image processing, and various image Transforms, Image Enhancement Techniques, Image restoration Techniques and methods, image compression and Segmentation used in digital image processing.

Learning Outcomes

Upon completion of this course, students will be familiar with basic image processing techniques for solving real problems. Student will also have sufficient expertise in both the theory of two-dimensional signal processing and its wide range of applications, for example, image restoration, image compression, and image analysis.

Course Synopsis

This course is an introduction to the fundamental concepts and techniques in basic digital image processing and their applications to solve real life problems. The topics covered include Digital Image Fundamentals, Image Transforms, Image Enhancement, Restoration and Compression, Morphological Image Processing, Nonlinear Image Processing, and Image Analysis. Application examples are also included.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction:** Digital Image Processing, Steps in Digital Image Processing, Components of image processing System, Image sensing and acquisition, sampling and quantization, relationships between pixels.	2
2.	**Image enhancement techniques:** Spatial domain, Frequency domain and using Fuzzy techniques. Intensity Transformation Functions, Histogram Processing, Fundamentals of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters. Filtering in the Frequency Domain.	2
3.	**Image Restoration and Reconstruction:** A Model of the Image Degradation/Restoration Process, Noise Models, Restoration in the Presence of Noise Only-Spatial Filtering, Periodic Noise Reduction by Frequency Domain Filtering Linear, Position-Invariant Degradations, Inverse Filtering, Wiener Filtering, Constrained Least Squares Filtering, Geometric Mean Filter.	2
4.	**Color Image Processing:** Color Models, Color Transformations, Image Segmentation Based on Color.	2
5.	**Wavelets and Multiresolution Processing:** Background, Multiresolution Expansions, Wavelet Transforms in One Dimension	2

6.	**Image Compression and Segmentation:** Fundamentals, Image Compression Models, Compression Methods, Point, Line, and Edge Detection, Thresholding, Region-Based Segmentation.	2
7.	**Pattern Recognition:** Introduction, importance, Features, Feature vectors, and classifiers,Supervised, unsupervised and semi-supervised Learning, Bayes Decision Theory, Bayesian classification for Normal Distributions, The Naïve - Bayes Classifier, The Nearest Neighbor Rule.	2
Text Books		
1.	Rafael C.Gonzalez and Richard E.Woods, "Digital Image Processing", Third Edition,Pearson Education.	
2.	SergiosTheodoridis, KonstantinosKoutroumbas, Pattern Recognition.	
References		
1.	Pratt, W. K, "Digital Image Processing".	
2.	Anil K. Jain, "Fundamentals of Digital Image Processing", Prentice-Hall India, 2007.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	System Design using HDL	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST013	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course instructs the students in the use of VHDL ((Very High Speed Integrated Circuit Hardware Description Language) for describing the behaviour of digital systems. VHDL is a standardized design language used in computer/ semiconductor industry. This course will teach students the use of the VHDL language for representation of digital signals, use of IEEE standard logic package/library, design description, design of arithmetic, combinational, and synchronous sequential circuits.

Learning Outcomes

- Learn the IEEE Standard 1076 Hardware Description Language (VHDL).
- Be able to model complex digital systems at several level of abstractions; behavioral and structural, synthesis and rapid system prototyping.
- Be able to develop and simulate register-level models of hierarchical digital systems.
- Develop a formal test bench from informal system requirements.
- Be able to design and model complex digital system independently or in a team.

Course Synopsis

Design and evaluation of control and data structures for digital systems. Hardware design languages are used to describe and design both behavioral and register transfer level architectures and control units with a microprogramming emphasis. Cover basic computer architecture, memories, digital interfacing, timing and synchronization, and microprocessor systems.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction:** VHDL description of combinational networks, Modeling flip-flops using VHDL, VHDL models for a multiplexer, Compilation and simulation of VHDL code, Modeling a sequential machine, Variables, Signals and constants, Arrays, VHDL operators, VHDL functions, VHDL procedures, Packages and libraries, VHDL model for a counter.	2
2.	**Designing With Programmable Logic Devices:** Read-only memories, Programmable logic arrays (PLAs), Programmable array logic (PLAs), Other sequential programmable logic devices (PLDs), Design of a keypad scanner.	2
3.	**Design Of Networks For Arithmetic Operations:** Design of a serial adder with accumulator, State graphs for control networks, Design of a binary multiplier, Multiplication of signed binary numbers, Design of a binary divider.	1
4.	**Digital Design with SM Charts:** State machine charts, Derivation of SM charts, Realization of SM charts. Implementation of the dice game, Alternative realization for SM charts using microprogramming, Linked state machines.	2
5.	**Designing With Programmable Gate Arrays And Complex Programmable Logic Devices:** Xlinx 3000 series FPGAs, Designing with FPGAs, Xlinx 4000 series FPGAs, using a one-	2

	hot state assignment, Altera complex programmable logic devices (CPLDs), Altera FELX 10K series COLDs.	
6.	**Floating - Point Arithmetic:** Representation of floating-point numbers, Floating-point multiplication, Other floating-point operations.	1
7.	**Additional Topics In VHDL:** Attributes, Transport and Inertial delays, Operator overloading, Multi-valued logic and signal resolution, IEEE-1164 standard logic, Generics, Generate statements, Synthesis of VHDL code, Synthesis examples, Files and Text IO.	2
8.	**VHDL Models For Memories And Buses:** Static RAM, A simplified 486 bus model, Interfacing memory to a microprocessor bus.	2
Text Books		
1.	Digital Systems Design Using VHDL by Charles H. Roth, Jr. and Lizy Kurian John, 2nd Edition, Thomson.	
2.	The Student's Guide to VHDL by Peter J. Ashenden, Morgan Kaufmann.	
References		
1.	'Fundamentals of Digital Logic with VHDL Design', by S. Brown and Z. Vranesic, Third edition, McGraw Hill, 2009.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Real Time Systems	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST014	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course includes:

- Abstract models of timed computation and the analysis of scheduling algorithms.
- Understand the motivation, theoretical background, and some of the work that has been done in the field of real-time systems.

Learning Outcomes

After completion of course students will be able to:

- Explain fundamental principles for programming of real time systems with time and resource limitations.
- Describe the foundation for programming languages developed for real time programming.
- Account for how real time operating systems are designed and functions.
- Use real time system programming languages and real time operating systems for real time applications.
- Analyse real time systems with regard to keeping time and resource restrictions.

Course Synopsis

This course includes: Theory, algorithmic and protocol concepts, mechanisms, and implementations of real-time computer systems; Real-time scheduling, real-time synchronization, real-time operating system kernels, and real-time programming languages; Design and analysis of real-time resource management algorithms, their implementations in production operating system kernels, and real-time application development.

Course Outline / Content

Unit	Topics	Week
1.	**Real Time Systems:** Concept of Real Time System, Performance measures of Real Time System, Real Time Application. Real time computing - Concepts; Structure of a real time system – Characterization of real time systems and tasks - Hard and Soft timing constraints - Issues in real time computing, Design Challenges - Performance metrics - Prediction of Execution Time : Source code analysis, Micro-architecture level analysis, Cache and pipeline issues- Programming Languages for Real-Time Systems.	3
2.	**Task Assignment and Scheduling:** Different task model, Scheduling hierarchy, Offline versus Online Scheduling, Clock Drives. Model of Real Time System,	2
3.	**Scheduling:** Hierarchy scheduling of Periodic Task - Assumptions, fixed versus dynamic priority algorithms, schedulability test for fixed priority task with arbitrary deadlines. Scheduling of A-periodic and Sporadic Tasks. Scheduling for	3

	applications having flexible constrains, Scheduling Real Time Tasks in Multiprocessor and Distributed Systems.	
4.	**Resources and Resource Access Control:** Handling Resource sharing and dependency among real time tasks - Assumptions on resources and their usage, resource contention, resource access control (Priority Ceiling Protocol, Priority Inheritance protocol, Slack Based Priority Ceiling Protocol, Pre-emption Ceiling Protocol).	3
5.	**Communication and Databases:** Real Time Communication (hard and soft real time communication, traffic scheduling disciplines, QoS guarantees), Real Time Databases (Optimistic vs Pessimistic concurrency control protocols).	3
Text Books		
1.	C.M. Krishna, Kang G. Shin, Real Time Systems, International Edition, McGraw Hill Companies.	
2.	Jane W.S. Liu, Real-Time Systems, Pearson Education India, 2000.	
References		
1.	Philip A. Laplante and Seppo J. Ovaska, "Real-Time Systems Design and Analysis: Tools for the Practitioner'' IV Edition IEEE Press, Wiley. 2011	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Unix & Shell Programming	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST015	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

To provide knowledge about Unix operating system working principles, its file system and programming for inter-process communication. It also gives an understanding for using various system calls.

Learning Outcomes

By the end of this course, the student will be able to:

- Develop text data processing applications using Unix commands and filters;
- Design and develop text based user interface components;
- Understand user management, network management and backup utilities.

Course Synopsis

Common commands; Permissions; Command line structure; Trapping exit codes; Catching interrupts; Unix system calls; Signal and Interrupts; Variables and error recovery.

Course Outline / Content

Unit	Topics	Week
1.	File and common commands - Shell - More about files- Directories- Unix system - Basics of file Directories and filenames - Permissions - modes - Directory hierarchy - Devices – the grep family - Other filters - the stream editor sed - the awk pattern scanning and processing language - files and good filters.	3
2.	Command line structure - Metacharacters - Creating new commands - Command arguments and parameters - program output as arguments – Shell variables - More on I/O redirection - loop in shell programs - Bundle - Setting shell attributes, Shift command line parameters - Exiting a command or the shell, evaluating arguments - Executing command without invoking a new process - Trapping exit codes – Conditional expressions.	3
3.	Customizing the cal command, Functions of command, While and Until loops - Traps - Catching interrupts - Replacing a file- Overwrite - Zap - Pick command – News command - Get and Put tracking file changes.	3
4.	Standard input and output – Program arguments - file access - A screen at a time printer - On bugs and debugging - Examples- Zap pick - Interactive file comparison program - Accessing the environment - Unix system calls – Low level I/O, File system Directories and modes, Processors, Signal and Interrupts.	2
5.	Program development - Four function calculator - Variables and error recovery – Arbitrary variable names, Built in functions, Compilation into a machine, Control flow and relational operators, Functions and procedures - Performance evaluation- Ms macro package – Troff level – Tbl and eqnpreprocessors- Manual page - Other document preparation.	3

Text Books

1.	The Design of the Unix Operating System, Maurice J. Bach, First Edition, Pearson Education, 1999.
2.	Unix and shell Programming Behrouz A. Forouzan, Richard F. Gilberg.Thomson.
3.	Your UNIX the ultimate guide, Sumitabha Das, TMH, 2nd Edition.
References	
1.	UNIX for programmers and users, 3rd edition, Graham Glass, King Ables, Pearson Education.
2.	UNIX programming environment, Kernighan and Pike, PHI. / Pearson Education.
3.	The Complete Reference UNIX, Rosen, Host, Klee, Farber, Rosinski, Second Edition, TMH.

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	High Speed Networks	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST016	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- To develop an understanding of the basics of high speed networking technologies.
- To apply the concepts learnt in this course to optimize performance of high-speed networks.

Learning Outcomes

After completion of this course, students should be able to:

- Demonstrate the knowledge of network planning and optimization
- Design and configure networks to support number of applications.

Course Synopsis

High speed networks; Frame Relay Networks; High Speed LANs; Queuing Models; TCP and ATM congestion control; protocols for QoS support.

Course Outline / Content

Unit	Topics	Week
1.	High speed networks, Frame Relay Networks, Asynchronous transfer mode, ATM Protocol Architecture, ATM logical Connection, ATM Cell, ATM Service Categories, AAL.	4
2.	High Speed LANs.	2
3.	Queuing Models, Single Server Queues, Effects of Congestion, Congestion Control, Traffic Management, Congestion Control in Packet Switching Networks, Frame Relay Congestion Control.	4
4.	TCP and ATM congestion control, Integrated and Differentiated services, Integrated services architecture approach, components, services, queuing, protocols for QoS support.	4

Text Books

1. Behrouz A. Forouzan, Data Communication and Networking, Third Edition, Tata McGraw-Hill 2003
2. William stallings, "ISDN and broadband ISDN with frame relay and ATM", Pearson Education Asia, Fourth Edition, 2001

References

1. Andrew S. Tanenbaum, Computer Networks, Fourth Edition, Prentice Hall India 2002
2. Tom Sheldon, Encyclopedia of Networking and Telecommunication, Tata McGraw Hill, 2001

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Advanced Algorithms	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST017	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course gives a broad yet deep exposure to algorithmic advances of the past few decades, and brings students up to a level where they can read and understand research papers in algorithms. Thematically, the biggest difference from undergrad algorithms is extensive use of ideas such as randomness, approximation, high dimensional geometry, which are increasingly important in most applications. We will encounter notions such as algorithm design in face of uncertainty, approaches to handle big data, handling intractability, heuristic approaches, etc.

Learning Outcomes

By the end of the course, the student must be able to:

- Use a suitable analysis method for any given algorithm
- Prove correctness and running-time bounds
- Design new algorithms for variations of problems studied in class
- Select appropriately an algorithmic paradigm for the problem at hand
- Define formally an algorithmic problem.

Course Synopsis

Algorithm analysis techniques: worst-case and amortized, average-case, randomized, competitive, approximation. Basic algorithm design techniques: greedy, iterative, incremental, divide-and-conquer, dynamic programming, randomization, linear programming. Examples from graph theory, linear algebra, geometry, operations research, and finance. Approximation Algorithms, Linear Programming, Optimization, P, NP Classes of Complexity.

Course Outline / Content

Unit	Topics	Week
1.	**Analysis of Algorithms :**Review of algorithmic strategies, asymptotic analysis: upper and lower complexity bounds. Identifying differences among best, average and worst Case Behaviours. Big O, little O, omega and theta notations, Standard complexity classes. Empirical measurements of performance. Time and space trade-offs in algorithms. Analysing recursive algorithms using recurrence relations.	3
2.	**Fundamental Computing Algorithms:** Numerical algorithms, Sequential and binary search algorithms. Quadratic sorting algorithms and O (n log n) sorting algorithms. Algorithms on graphs and their complexities using Greedy Approach for – Prim's and Kruskal's Algorithm for minimum spanning tree, Single source shortest path Algorithm, all pair shortest paths in Graph – Bellman Ford Algorithm, Floyd Warshall Algorithm.	2
3.	**Approximation Algorithms:** Introduction, Approximation Algorithms for – Vertex Cover, Sum of Subsets, TSP, Job scheduling, Knapsack Problems. Probabilistically good algorithms, Polynomial Time Approximation.	3

4.	**Linear Programming :**Introduction, initial basic feasible solution. Feasibility of a system, Simplex Algorithm. Standard and Slack forms, Formulation of problems as linear programs, Checking Feasibility of System using B – Rule Algorithm. Optimization. KKT Algorithm. Expectations: Introduction, Moments, Expectations of functions of more than one random variable.	4
5.	**Computational complexity:** Complexity measures, Polynomial versus non-polynomial time complexity; NP hard and NP complete classes.	2
Text Books		
1.	Kishore S. Trivedi, "Probability & Statistics with Reliability, Queuing, and Computer Science Applications" PHI	
2.	Cormen, Leiserson, Rivest, "Algorithms", PHI	
3.	Bressard, "Fundamentals of Algorithms", PHI	
References		
1.	Steven S Skiena, "The Algorithm Design Manual" – Springer Publications	
2.	Knuth, " The Art of Programming", Addison Wesley Vol I and II	
3.	Michael T Goodrich, "Algorithm Design" WILEY Publications.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Reconfigurable Computing	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST018	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

Learning Outcomes

Students will gain fundamental knowledge and understanding of principles and practice in digital design on FPGAs through class lectures. Students will also learn the programming in VHDL.

Course Synopsis

Reconfigurable Computing Hardware; Programming Reconfigurable Systems; Mapping; Designs to Reconfigurable Platforms; Application Development: CORDIC Architectures for; FPGA Computing; FPGA Applications.

Course Outline / Content

Unit	Topics	Week
1.	**Reconfigurable Computing Hardware:** Device Architecture, Reconfigurable Computing Architectures**,** Reconfigurable Computing Systems,Reconfiguration Management.	2
2.	**Programming Reconfigurable Systems:** Compute Models and System Architectures, Programming FPGA Applications in VHDL, Compiling C for Spatial Computing, Stream Computations Organized for Reconfigurable Execution, Programming Data Parallel FPGA Applications Using the SIMD/Vector Model, Operating System Support for Reconfigurable Computing.	3
3.	**Mapping Designs to Reconfigurable Platforms:** Technology Mapping, FPGA Placement Placement for General-purpose FPGAs, Data-path Composition, Specifying Circuit Layout on FPGAs, Retiming, Re-pipelining, and *C*-slow Retiming, Configuration Bit-stream Generation, Fast Compilation Techniques	3
4.	**Application Development:** Implementing Applications with FPGAs, Instance-specific Design, Precision Analysis for Fixed-point Computation, Distributed Arithmetic, CORDIC Architectures for FPGA Computing, Hardware/Software Partitioning	3
5.	**Case Studies of FPGA Applications**: SPIHT Image Compression, Automatic Target Recognition Systems on Reconfigurable Devices, Boolean Satisfiability: Creating Solvers Optimized for Specific Problem Instances, Multi-FPGA Systems: Logic Emulation, Finite Difference Time Domain: A	3

	Case Study Using FPGAs, Network Packet Processing in Reconfigurable Hardware	
Text Books		
1.	Scott Hauck and Andre DeHon, "Reconfigurable Computing – The Theory and Practice of FPGA-based Computation", ELSEVIER 2008	
References		
1.	Christophe Bobda "Introduction to Reconfigurable Computing: Architectures, Algorithms, and Applications" SPRINGER 2007.	
2.	JariNurmi, "Processor Design: System-On-Chip Computing for ASICs and FPGAs". SPRINGER 2008.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Computer Vision	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST019	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

To introduce students the fundamentals of image formation; To introduce students the major ideas, methods, and techniques of computer vision and pattern recognition; To develop an appreciation for various issues in the design of computer vision and object recognition systems; and To provide the student with programming experience from implementing computer vision and object recognition applications.

Learning Outcomes

After completing the course you will be able to:

- Identify basic concepts, terminology, theories, models and methods in the field of computer vision.
- Describe known principles of human visual system.
- Describe basic methods of computer vision related to multi-scale representation, edge detection and detection of other primitives, stereo, motion and object recognition.
- Suggest a design of a computer vision system for a specific problem.

Course Synopsis

Computer Vision plays a very important role in fields such as Machine and Robot Intelligence. They provide the means for the machine or robot to interact intelligently with the outside world through visual perception. Vision is undoubtedly the most powerful of all senses and enables robots to perform very flexible tasks such as moving around autonomously in a factory floor or outdoors. The applications are plentiful and very challenging. Face recognition, human activity interpretations, human-computer interaction, quality inspection of mass-produced parts, robot/missile/vehicle guidance, medical imaging and computer vision-aided surgery are some of the applications. The objective of this course is to prepare students for working in such intelligent automation fields.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction:** History about computer vision, introduction to vision, computer graphics, image processing, human and computer vision.	1
2.	**Image Formation Models:**Monocular imaging system, Orthographic & Perspective Projection, Camera model and Camera calibration, Binocular imaging systems.	1
3.	**Recognition Methodology:** Conditioning, Labeling, Grouping, Extracting and Matching.	1
4.	**Morphological Image Processing:** Introduction, Dilation, Erosion, Opening, Closing, Hit-or-Miss transformation, Morphological algorithm operations on binary images, Morphological algorithm Operations on gray-scale images, Thinning, Thickening, Region growing, region shrinking.	1
5.	**Image Representation and Description:** Representation schemes, Boundary descriptors, Region descriptors.	1
6.	**Binary Machine Vision:**Thresholding, Segmentation, Connected component labeling, Hierarchal segmentation, spatial	2

	clustering, Split & merge, Rule-based Segmentation, Motion-based segmentation.	
7.	**Area Extraction:** Concepts, Data-structures, Edge, Line-Linking, Hough transform, Line fitting, Curve fitting (Least-square fitting).	
8.	**Region Analysis:** Region properties, External points, spatial moments, mixed spatial gray-level moments, Boundary analysis: Signature properties, Shape numbers.	
9.	**Facet Model Recognition:**Labeling lines, Understanding line drawings, Classification of shapes by labeling of edges, Recognition of shapes, Consisting labeling problem, Back-tracking Algorithm Perspective Projective geometry, Inverse perspective Projection, Photogrammetry - from 2D to 3D.	2
10.	**Image matching:** Intensity matching of ID signals, Matching of 2D image, Hierarchical image matching, 2D representation, Global vs. Local features. **General Frame Works for Matching:** Distance relational approach, ordered structural matching, View class matching, Models database organization.	2
11.	**General Frame Works:** Distance -relational approach, Ordered -Structural matching, View class matching, Models database organization.	1
12.	**Knowledge Based Vision:** Knowledge representation, Control strategies, Information Integration.	1
13.	**Object recognition:**Hough transforms and other simple object recognition methods**,** Shape correspondence and shape matching Principal component analysis**,** Shape priors for recognition	1
Text Books		
1.	"Computer and Robot Vision", Robert Haralick and Linda Shapiro, Addison Wesley.	
2.	"Computer Vision: A Modern Approach", David A. Forsyth, Jean Ponce.	
3.	“Introductory Techniques for 3D Computer Vision”, E.Trucco and A.Verri, PHI.	
References		
1.	"Image Processing, Analysis, and Machine Vision", Milan Sonka, Vaclav Hlavac, Roger Boyle, Thomson Learning.	
2.	“Robot Vision”, by B. K. P. Horn, McGraw-Hill.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Advanced Computer Network	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST020	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

The objective for this course is to give students hands-on exposure to emerging networking topics in the context of a number of different tools/environments that might be used for networking research.

Learning Outcomes

In general terms, the course will deliver the following learning outcomes:

- To identify and discuss the concepts underlying IPv6 protocol, and their main characteristics and functionality;
- To understand the principles and functionality of mobile IP, explaining its concretization in IPv6; to understand the needs of optimization of the mobility mechanisms and description of some extensions that aim to reduce handover latency and requirements from terminals;
- To explain and exemplify current QoS architectures and mechanisms, and the QoS support challenges in future networks;
- To understand and explain the design issues in transport services in face of applications and services requirements;
- To understand theoretical and practical concepts behind the design of multiconstained applications and services;
- To discuss relevant management issues and devise adequate network management solutions;
- To identify and assess possible research opportunities and difficulties within the course scope.

Course Synopsis

This course will focus on advanced networking topics by studying a combination of classic research papers as well as current and emerging topics in computer networking and by doing a number of hands-on lab assignments. Specific focus areas will include cloud computing, network management, network measurement, software defined networking and network architectures. As such the course is suitable for Masters and PhD students wishing to explore or engage in networking related research.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction to Computer Networks** Review – Computer networks and layered architecture. Asynchronous Transfer Mode: ATM layered model, switching and switching fabrics, network layer in ATM, QOS, and LAN emulation.	2
2.	**Transport Layer** Elements of transport protocols; Internet transport protocols: TCP and UDP, TCP connection management, congestion control.	2
3.	**Application Layer** Network application architectures: Client-server, P2P and hybrid;	3

	Application layer protocols: DNS, FTP, TFTP, TELNET, HTTP and WWW, SMTP and electronic mail; Network management and SNMP.	
4.	**Wireless and Mobile:** Wireless and Mobile Networks: Wireless links and network characteristics, 802.11 wireless LANs, mobility management, addressing and routing, mobile IP, WAP, mobility in cellular networks.	3
5.	**Multimedia Networking:** Streaming audio and video, RTSP, jitter removal and recovery from lost packets; Protocols for real-time interactive applications: RTP, RTCP, SIP, H.323; Content distribution networks; Integrated and differentiated services, RSVP.	2
6.	**Introduction to Network Security** Cryptography, symmetric and public-key algorithms, digital signatures, communication security, authentication protocols, E-mail security, PGP and PEM.	2
Text Books		
1.	Kurose, J. F. and Ross, R.W,Computer Networking, Pearson Education	
References		
1.	Comer, D.E. and Droms, R.E, Computer Networks and Internets, Prentice-Hall.	
2.	Walrand, J. and Varaiya, P, High Performance Communication Networks, Morgan Kaufmann.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Advanced Computer Graphics	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST021	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course is a computer graphics class at the graduate level. The course mainly consists of lectures covering recent research results, ranging from mesh processing, simulation, to non-photorealistic rendering necessary basic mathematical and computation tools will be introduced when needed. Everyone will be expected to complete one or two individual project(s), present one paper related to a chosen research topic (as a team), and complete a (team) project.

Learning Outcomes

Students completing this course are expected to be able to:

- Understand the basics of geometry processing.
- Understand and work with advanced rendering methods such as radiosity.
- Design programs for advanced animation methods.
- Understand issues of modern graphics research.

Course Synopsis

This course covers advanced topics in computer graphics. We will focus on two specific questions: How to create photo-realistic renderings and how to create physically plausible animations? To answer the first question, we will first discuss and analyze the classical raytracing algorithm. With an understanding of the limitations of raytracing, we will look at a more principled way of image synthesis based on the physics of light transport. After studying the basic physical quantities of light transport and corresponding local illumination models, we will derive the global rendering equation as a model for image synthesis. We then discuss Monte Carlo methods for evaluating this integral equation leading to several Monte Carlo rendering algorithms such as path tracing or photon mapping. In the second part of the course we will study concepts and algorithms for the animation of solids and fluids, and discuss principles of performance-driven character animation. Starting with simple particle systems and mass-spring networks, we will discuss numerical time integration methods commonly applied for computer animation. Rigid body simulation and elastic materials will also be covered. We then look at how the approximate solutions of the Navier-Stokes equations can be computed to simulate fluid flow. Finally, we study advanced methods for animating 3D characters based on recorded performances.

Course Outline / Content

Unit	Topics	Week
1.	**Advanced Rendering Techniques:** Photorealistic rendering, Global Illumination, Participating media rendering, Ray tracing, Monte Carlo algorithm, Photon mapping.	3
2.	**Texture Synthesis and Image Processing:** Environmental mapping, Texture synthesis, anisotropic image smoothing.	2
3.	**Volume Rendering:** Volume graphics overview, Marching cubes, Direct volume rendering.	3
4.	**Surfaces and Meshes:** Subdivision, Distance fields and level sets.	2
5.	**Physically-based Modeling:** Stable fluid solver, Lattice Boltzmann method.	2

6.	**Individual Project**	2
Text Books		
1.	James D. Foley, Andries van Dam, Steven K. FeinerandJohn F. Hughes, Computer Graphics: Principles & Practices, Addison Wesley, 2nd edition in C, 1995.	
2.	Alan H. Watt and Mark Watt, Advanced Animation and Rendering Techniques: Theory and Practice, Addison-Wesley, 1992.	
References		
1.	Matt Pharr and Greg Humphreys, Physically based rendering, Morgan Kaufmann, 2004	
2.	Tomas Moller and Eric Haines Real-Time Rendering A K Peters Ltd, 2nd edition, 2002.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Advanced Database Management Systems	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST022	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

Effective collection, analysis, and maintenance of data is key to achieve rapid progress in almost all disciplines of science and engineering. In this course, we will cover the core principles and techniques of data and information management. The potential topics covered in class include processing and optimization of declarative queries, transactions, crash recovery, data stream systems, Advanced Application Development, Web data management (e.g., Internet and intranet search engines), information integration (e.g., semi structured data and XML), and data mining.

Learning Outcomes

- Master the basic concepts and appreciate the applications of database systems.
- Master the basics of SQL and construct queries using SQL.
- Be familiar with a commercial relational database system (Oracle) by writing SQL using the system.
- Be familiar with the relational database theory, and be able to write relational algebra expressions for queries.
- Mater design principles for logical design of databases, including the E-R method and normalization approach.
- Be familiar with basic database storage structures and access techniques: file and page organizations, indexing methods including B-tree, and hashing.
- Master the basics of query evaluation techniques query optimization.
- Be familiar with the basic issues of transaction processing and concurrency control.

Course Synopsis

Network, hierarchical, and relational, and entity-relationship models; data definition, manipulation languages, and conversion among these models; relational database design theory, efficient query evaluation, elementary query optimization techniques. Semi-Structured Data, Introduction to XML. Performance Tuning, Performance Benchmarks, Standardization, E-Commerce, HADOOP.

Course Outline / Content

Unit	Topics	Week
1.	**Physical Database Design & Tuning:** Database workloads, physical design and tuning decisions, Need for Tuning Index selection: Guideline for index selection, Clustering & Indexing Tools for index selection Database Tuning: Tuning indexes, Tuning Conceptual schema Tuning Queries &views, Impact of Concurrency, Benchmarking.	3
2.	**Advanced Transaction Processing:** Transaction Processing Monitors, Transactional Workflow, Real time transaction System, Long duration Transactions, Transaction Management in Multi-databases, Distributed Transaction Management, Main Memory Databases, and Advanced Transaction Models.	2
3.	**Semi-Structured Data and XML:** Semi-Structured Data,	4

	Introduction to XML, XML hierarchical Model, DTD & XML schema, XML Namespace, XML query & Transformation: Xpath, XSLT, XQuery, Storage of XML data, XML Technologies: DOM &SAX Interfaces X pointer, Xlink, XHTML, SOAP, WSDL, UDDI, XML database Application.	
4.	**Emerging Trends in Databases:** Introduction, Motivation, Temporal databases, Spatial & geographic databases, Multimedia Databases, Mobility & personal Databases.	2
5.	**Advanced Application Development:** Performance Tuning, Performance Benchmarks, Standardization, E-Commerce, Legacy Systems, Large-scale Data Management with HADOOP, Semi structured database COUCHDB: Introduction, Architecture and principles, features.	3
Text Books		
1.	Database system Concept by Silberschatz and Korth 6th Edition	
2.	Distributed Databases principles & systems by Stefano Ceri, Giuseppe Pelagatti	
3.	Database Systems, Thomas Connolly, Carolyn Begg, Pearson 4th Edition	
References		
1.	Web Data Management, Abiteboul, Loana, Philippe et.al Cambridge publication.	
2.	Database Management Systems by Raghu Ramakrishnan and Johannes Gehrke	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Advanced Computer Architecture	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST023	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

Basic understanding of Computer Architecture, Multi-Threading and Multi-Core programming concepts. The student should be made to: Understand the micro-architectural design of processors. Learn about the various techniques used to obtain performance improvement and power savings in current processors

Learning Outcomes

At the end of the course, the student should be able to: Evaluate performance of different architectures with respect to various parameters Analyze performance of different ILP techniques Identify cache and memory related issues in multi-processors

Course Synopsis

An overview of computer architecture, which stresses the underlying design principles and the impact of these principles on computer performance. General topics include Thread Level Parallelism & Multi-Core Architecture, memory organization, system organization, thread level parallel processing and Multi-Core Programming.

Course Outline / Content

Unit	Topics	Week
1.	**Modern Computer Architectures** Introduction, Fundamentals of RISC, CISC, Instruction Level Parallelism(ILP) – Concepts and Challenges, Branching with Prediction, Dynamic Scheduling: Hazards and Solutions, Measuring Performance of ILP, Limitations of ILP.	2
2.	**Thread Level Parallelism & Multi-Core Architecture** Thread Level Parallelism, Simultaneous Multi-Threading, Multi-Processor Architecture: Types, Limitations; Evolution of Multi-Core, Architecting with Multi-Core: Homogenous and heterogeneous cores, Shared recourses, shared busses, and optimal resource sharing strategies, Performance Evaluation of Multi-Core Processors.	3
3.	**Memory Module Design** Conceptual view of memory cell, Memory address map, Memory connections to CPU, Cache memory-Cache memory management techniques. Types of cache's: Look through, look aside, write through, write around, unified Vs Split, multilevel, cache levels. Shared memory multiprocessors, Synchronization, small-scale symmetric multiprocessors on a snoopy bus, cache coherence on snoopy buses, Scalable multiprocessors, Directory-based cache coherence, Interconnection network, Memory consistency models, Software distributed shared memory.	3
4.	**Multi-Threading Concepts** Fundamentals of Multi-threaded programming, Concurrency vs Parallelism, Threading design concepts for developing an	3

	application, Correctness Concepts: Critical Region, Mutual exclusion, Synchronization, Race Conditions. Performance Concepts: Simple Speedup, Computing Speedup, Efficiency, Granularity, Load Balance. Multithreading in hardware, Chip multiprocessing, current research and future trends.	
5.	**Multi-Core Programming** Introduction to OpenMP, OpenMP directives, Parallel constructs, Work-sharing constructs, Data environment constructs, Synchronization constructs, Extensive API library for finer control, benchmarking multi-core architecture: Bench marking of processors. Comparison of processor performance for specific application domains.	3
Text Books		
1.	John L. Hennessy and David A. Patterson – Quantative Approach – Computer Architecture 5th edition,Morgan Kaufmann, 2011.	
2.	Shameem Akhter and Jason Roberts, ―Multi-Core Programming, 1st edition, Intel Press, 2006.	
References		
1.	Vincent. P. Heuring, Harry F. Jordan ―Computer System design and Architecture 2nd edition,Pearson, 2003.	
2.	Apman, Gabriele Jost, Ruud van van der Pas, ―Using OpenMP: Portable Shared Memory ParallelProgramming (Scientific and Engineering Computation), 1st edition, MIT Press, 2007.	
3.	H. J. Siegel.Interconnection Network for Large Scale Parallel Processing, McGraw Hill, 1990.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Advanced Compilation Techniques.	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST024	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

Built upon basic compiler knowledge, this course covers compiler architecture and techniques, including control flow analysis, optimization, pipelined architecture, garbage collection etc.

Learning Outcomes

After completion of this course, students will be able to:

- Understanding of the challenges involved in compilation (semantic gap between input and output languages, compiler efficiency and code quality)
- Understanding of the phases involved in compilation, and knowledge of the techniques applied.
- Ability to understand design decisions in modern compilers and to justify these.
- Ability to develop and apply modifications to standard compilation techniques wherever this is necessary.
- Ability to analyse compilation tasks and to apply standard compilation techniques.

Course Synopsis

This course includes the basic concepts related to compiler, its architecture , Control Flow Analysis, Static-single assignment, Scalar optimization, Instruction scheduling, Performance evaluation, Data dependence analysis, Loop transformations, Garbage collection and Advanced Topics.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction**: Compiler structure, architecture and compilation, sources of improvement.	2
2.	**Control flow analysis:** Basic blocks & loops. Data flow analysis and optimizations: bit vectors, iterative frameworks, interval analysis, reaching definitions, liveness, common subexpression elimination, constant propagation. More control flow analysis: dominators, control dependence.	2
3.	**Static-single assignment:** Static-single assignment, constant propagation. Scalar optimization: loop invariant code motion, common subexpression elimination, strength reduction, dead code elimination, loop optimizations, etc.	2
4.	**Instruction scheduling:** Pipelined architectures, delayed-load architectures, list scheduling. Register allocation: coloring, allocation, live range splitting.	2
5.	**Performance evaluation:** Interprocedural analysis: side effects, flow-insensitive, flow-sensitive, constants, inlining. Alias analysis: alias analysis, method resolution. Searching, indexing, and their implications to memory management. Information extraction and feature selection. Points-to Analysis Supervised, unsupervised-learning, and stream mining.	2

6.	**Data dependence analysis:** Dependence testing, dependence graphs.	1
7.	**Loop transformations:** Interchange, tiling, fusion, distribution, splitting Just-in-time compilation: fast global optimization.	1
8.	**Garbage collection:** Automatic memory management and data locality.Optimal Integrated Code Generation with OPTIMIST	2
Text Books		
1.	Compilers by Alfred V. Aho , Monica S Lam, R Sethi, Jeffrey D. Ullman	
2.	Mapping and Compilation: Methods and Techniques by K. K. Rampal	
References		
1.	https://www.ece.cmu.edu/~ece447/s13/lib/exe/fetch.php?...advancedcaching...ppt	
2.	David Bacon, Susan Graham, Oliver Sharp: Compiler Transformations for High-Performance Computing. ACM Computing Surveys, December 1994, Volume 26 Issue 4. Preprint	
3.	David A. Padua and Michael J. Wolfe: Advanced compiler optimizations for supercomputers.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Principles of Cryptography	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST025	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course is intended to provide a theoretically sound foundation in cryptography as used in network security. We shall learn about basic cryptographic tool like encryption and message authentication, in the "private-key" and "public-key" settings, with a focus on mathematical definitions of security, "provably secure" constructions based on fundamental cryptographic primitives, and how they are used in higher-level network security protocols.

Learning Outcomes

To provide a basic introduction to central aspects of symmetric and asymmetric cryptography. To establish knowledge and understanding of how cryptographic techniques are used to establish security in modern information- and communication systems.

Course Synopsis

Cryptography provides important tools for ensuring the privacy, authenticity, and integrity of the increasingly sensitive information involved in modern digital systems. Nowadays, core cryptographic tools, including encryption, message authentication codes, digital signature, key agreement protocols, etc., are used behind millions of daily on-line transactions. In this course, we will unveil some of the "magic" of cryptography. Modern Cryptography uses mathematical language to precisely pin down elusive security goals, design primitives and protocols to achieve these goals, and validate the security of designed primitives and protocols using mathematical proofs based on clearly stated hardness assumptions. Therefore, to learn cryptography, it is essential to understand its mathematical underpinning. In this course, students will see the inner-working of cryptography for several core cryptographic tools, from encryption, to message authentication codes, to hash functions, to digital signatures, etc.

Course Outline / Content

Unit	Topics	Week
1.	Introduction to Cryptography: Basics of Symmetric Key Cryptography, Basics of Asymmetric Key Cryptography, Hardness of Functions.	2
2.	Mathematical Background for Cryptography: Number Theory, GCD, Groups, Rings, Fields, Properties, Chinese Remainder Theorem.	2
3.	Classical Cryptography: Introduction to Some simple cryptosystems and their Cryptanalysis. Shannon's Theory, Secret vs. Public Key Cryptography.	2
4.	Stream ciphers and Block Ciphers: DES and Alternatives, AES.	2
5.	Cryptographic Hash Functions and MAC.	2
6.	Public Key Cryptography, RSA Cryptosystem and Factoring Integers.	

		2
7.	Discrete Logarithm Problem in Prime Fields, Generalized Discrete Logarithm Problem. Attacks against Discrete Logarithm Problem. Public Key Cryptosystems based on the Discrete Logarithm Problem.	2
8	Elliptic Curve Cryptosystems. Digital Signatures.	1
Text Books		
1.	Hans Delfs, Helmut Knebl, "Introduction to Cryptography, Principles and Applications", Springer Verlag.	
2.	Wenbo Mao, "Modern Cryptography, Theory and Practice", Pearson Education	
References		
1.	A Graduate Course in Applied Cryptography by Dan Boneh and Victor Shoup	
2.	Introduction to Modern Cryptography (2nd edition) by Jonathan Katz and Yehuda Lindell	
3.	Handbook of Applied Cryptography by A. Menezes, P. Van Oorschot, S. Vanstone.	
4.	O. Goldreich, Foundations of Cryptography, CRC Press.	

Department of Computer Science & Engineering

National Institute of Technology Srinagar

Course Title	Neural Networks	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST026	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- To introduce major deep learning algorithms, the problem settings, and their applications to solve real world problems.
- To understand the role of neural networks in engineering, artificial intelligence, and cognitive modelling.
- To provide knowledge in developing the different algorithms for neural networks

Learning Outcomes

After completion of this course the students should be able to:

- Identify the deep learning algorithms which are more appropriate for various types of learning tasks in various domains.
- Implement deep learning algorithms and solve real-world problems.

Course Synopsis

Basics of ANN-Mathematical model-Applications; Single Layer Perceptron; Multi Layer Perceptron; Associative Memory; Recurrent neural networks; Boltzmann machine; Self-organizing feature maps; Fuzzy neural networks,Genetic algorithms.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction to neural networks**: Biological and Artificial neurons,McCulloch Pitts Neuron, Thresholding Logic, Perceptrons, Perceptron Learning Algorithm and Convergence, Multilayer Perceptrons (MLPs), Representation Power of MLPs, Network Architectures.	2
2.	**Multilayer networks:**Sigmoid Neurons, Gradient Descent(GD) , Feedforward Neural Networks, Representation Power of Feedforward Neural Networks,Back propagation (BP) , Momentum Based GD, Nesterov Accelerated GD, Stochastic	3

	GD, Recurrent networks and unsupervised learning, Hopfield network - energy; stability; capacity,Boltzmann machine, Kohonen'sself organizing feature maps.	
3.	**Associative memory:**Auto associative memories, Hetero associative memories, performance measures, associative memory models, Applications of associative memories.	2
4.	**Convolutional Neural Networks:**Visualizing Convolutional Neural Networks, Guided Backpropagation**,**Fooling Convolutional Neural Networks.	2
5.	**Neuro Evolution:** Introduction to Neuro evolution, Weight evolution, Topology evolution, Learning rule evolution, Deep Neuro Evolution. Neuro evolution uses and introduction to evolutionary algorithms to build Neural networks	3
Text Books		
1.	Limin Fu." Neural Network in Computer Intelligence" , Tata McGraw-Hill 2003 Edition.	
2.	Gene Sher, "handbook of Neuro evolution" , Springer , Edition 1.	
References		
1.	James A. Freeman David M. Skapura, "Neural Networks: Algorithms, Applications, And Programming Techniques". Pearson Publication ,Edition 1	
2.	Ke-lin du, M.N.S Swamy, "Neural networks and statistical learning", Springer 2014 edition.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Pervasive Computing	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST027	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

The course is about the emerging discipline of Pervasive Computing, also known as Ubiquitous Computing, Everywhere Computing and Invisible Computing. The key element here is the omnipresence of information devices. These devices can be embedded into cars, airplanes, ships, bikes, posters, signboards, walls and even clothes. This course focuses on independent information devices including mobile phones, smart phones, and laptops (PCs), and the services made available by them. It includes human-computer interaction using several types of elements including text, speech, and handwriting.

Learning Outcomes

The objective of this course is:

- To study the pervasive computing and its applications.
- To study the pervasive computing web based applications.
- To study voice enabling pervasive computing.
- To study PDA in pervasive computing.
- To study user interface issues in pervasive computing.

Course Synopsis

The course aims at providing a sound conceptual foundation in the area of Pervasive Computing aspects. The course attempts to provide a balanced treatment of the mechanisms and environments of pervasive computing and initiates senior CS students to the state-of-the-art in the area. At the end of this course, students should be able to conceptualize, analyze and design select classes of pervasive computing systems.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction to Pervasive Computing:** Past, present, future; the pervasive computing market, m-Business, Challenges and future of Pervasive Computing, **Application Examples of Pervasive Computing:** Retail, Airline Check-in and booking, Sales force automation, Healthcare, Tracking, Car Information Systems, Email Access via WAP and voice	3
2.	**DeviceTechnology for PervasiveComputing :** Hardware, Human-machine interfaces, Biometrics, Operating Systems, Java for pervasivedevices, Outlook **Device Connectivity:** Protocols, Security, Device Management	3
3.	**Developing WML Applications:** Developing WML Applications: WML documents, developing a WML application, WML tags, registration WML listing and WML script.	2
4.	**MIDP Programming:** MIDP Programming: J2ME MIDP user interface, MIDP application, developing a MIDP application, MIDP Classes: MIDlet class, MIDP GUI classes, MIDlet high-level events, low-	2

	level APIs and event handling.	
5.	**Advanced MIDP Programming** : Network programming, MIDP database programming, MIDlet provisioning, Bluetooth application.	2
6.	**Developing VoiceXML Applications** Developing VoiceXML Applications: VoiceXML applications, VoiceXML TAGS, ECMAScript – Java Card Application: Java card VM, APDUs, java card API, host applications.	2
Text Books		
1.	JochenBurkhardt, Dr. Horst Henn, Stefan Hepper – Pervasive Computing Technology and Architecture ofMobile Internet Applications – Pearson Education, 2005.	
2.	JochenBurkhardt, Horst Henn, Stefan Hepper, Thomas Schaec& Klaus Rindtorff: Pervasive Computing: Technology and Architecture of Mobile Internet Applications, Pearson Education, New Delhi, 2006.	
References		
1.	Dan Harkey, Shan Appajodu, Mike Larkin – Wireless Java Programming for Enterprise Applications WileyPublishing, Inc., 2001.	
2.	StefenPoslad: Ubiquitous Computing: Smart Devices, Environments and Interactions, Wiley, Student Edition, 2010.	
3.	A. Genco, S. Sorce: Pervasive Systems and Ubiquitous Computing, WIT Press, 2012.	
4.	Guruduth S. Banavar, Norman H. Cohen, ChandraNarayanaswami: Pervasive Computing: An Application-Based Approach, Wiley Interscience, 2012.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Distributed and Parallel Computing	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST028	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

To provide knowledge on principles and practice underlying in the design of distributed systems.

Learning Outcomes

By the end of the course, the students will be able to:

- Layout foundations of Distributed Systems;
- Introduce the idea of middleware and related issues;
- Understand in detail the system level and support required for distributed system;
- Understand the issues involved in studying data and design of distributed algorithms.

Course Synopsis

overview of parallel computing; Pipelined computations; communication technologies; clock synchronization; proof of correctness; complexity analysis; Distributed operating systems; algorithms for implementing DSM; load balancing; fault-tolerant models; Research issues in distributed systems.

Course Outline / Content

Unit	Topics	Week
1.	An overview of parallel computing, Languages and programming environments, Message passing computing, Partitioning and divide-and-conquer strategies, Pipelined computations, Synchronous computations, Load balancing and termination detection, Programming with shared memory.	2
2.	Algorithms and applications Components of distributed systems, Communication technologies, communication services.	1
3.	Distributed algorithms and protocols: examples of distributed algorithms, clock synchronization, logical and vector clocks, election algorithms, consensus algorithms, proof of correctness, complexity analysis.	2
4.	Distributed operating systems: system models, file services, name services, process synchronization and coordination, case studies.	2
5.	Distributed shared memory: algorithms for implementing DSM, coherence protocols.	1
6.	Distributed resource management: load sharing, load balancing, resource monitoring	2
7.	Failure recovery and fault tolerance: check-pointing, recovery, fault-tolerant models and protocols	2
8.	Research issues in distributed systems, real-time protocols, standardization issues, cluster and grid computing.	2

Text Books

1. George Coulouris, Jean Dellimore and Tim KIndberg, “Distributed Systems Concepts and Design”, Pearson Education.

2.	Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet (DCC) by Kai Hwang, Jack Dongarra& Geoffrey C. Fox.
3.	Ajay D. Kshemkalyani and MukeshSinghal, "Distributed Computing – Principles.
4.	Andrew S. Tanenbaum and Maarten van Steen. "Distributed Systems: Principles and Paradigms" (DSPD), Prentice Hall
2.	Principles of Parallel Programming, by Calvin Lin and Larry Snyder, Addison-Wesley.
References	
1.	MukeshSinghal and N. G. Shivaratri, "Advanced Concepts in Operating Systems"
2.	Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, by Barry Wilkinson, Michael Allen. Prentice Hall.
3.	Joshy Joseph and Craig Fellenstein, "Grid Computing", IBM Press.
4.	Algorithms and Systems", Cambridge University Press
5.	Nancy A. Lynch, Distributed Algorithms, Morgan Kaufmann Publishers.

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Cloud Computing	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST029	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course will introduce various aspects of cloud computing, including fundamentals, management issues, security challenges and future research trends. This will help students (both UG and PG levels) and researchers to use and explore the cloud computing platforms.

Learning Outcomes

This course offers a good understanding of cloud computing concepts and prepares students to be in a position to design cloud based applications for distributed systems.

Course Synopsis

The course presents a top-down view of cloud computing, from applications and administration to programming and infrastructure. Its main focus is on parallel programming techniques for cloud computing and large scale distributed systems which form the cloud infrastructure. The topics include: overview of cloud computing, cloud systems, parallel processing in the cloud, distributed storage systems, virtualization, security in the cloud, and multicore operating systems. Students will study state-of-the-art solutions for cloud computing developed by Google, Amazon, Microsoft, Yahoo, VMWare, etc. Students will also apply what they learn in one programming assignment and one project executed over Amazon Web Services.

Course Outline / Content

Unit	Topics	Week
1.	**Cloud Computing Basics:** Cloud Computing Overview; Characteristics; Applications; Internet and Cloud; Benefits; Limitations; Challenges.	1
2.	**Cloud Computing Services and Deployment Models:** Infrastructure as a Service; Platform as a Service; Software as a Service; Private Cloud; Public Cloud; Community Cloud; Hybrid Cloud	2
3.	**Cloud Computing vs Other Computing Technologies:** Overview of Grid, Peer-to-Peer, Pervasive and Utility Computing technologies; their characteristics and comparison between them	1
4.	**Accessing the Cloud:** Hardware and Infrastructure requirements; Access Mechanisms: Web Applications, Web APIs, Web Browsers.	2
5.	**Cloud Storage and Cloud Standards:** Overview; Storage as a Service; Cloud Storage Issues; Challenges; Standards	2
6.	**Security Issues**: Securing the Cloud, Securing Data, Establishing identity and presence.	2
7.	**Developing Applications:** Major Players in Cloud Business; Overview of Service Oriented Architecture; Tools for developing cloud services and	2

	applications.	
8.	**Practice Cloud IT Model:** Analysis of Case Studies when deciding to adopt cloud computing architecture. How to decide if the cloud is right for your requirements. Cloud based service, applications and development platform deployment so as to improve the total cost of ownership (TCO).	2
Text Books		
1.	Anthony T. Velte, Toby J. Velte, and Robert Elsenpeter: Cloud Computing: A Practical Approach, McGraw Hill, 2010.	
2.	Kai Hwang, Jack Dongarra& Geoffrey C. Fox.:Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet (DCC)	
References		
1.	RajkumarBuyys, James Broberg, AndrzejGoscinski (Editors) : Cloud Computing: Principles and Paradigms, Wiley, 2011.	
2.	Barrie Sosinsky : Cloud Computing Bible, Wiley, 2011.	
3.	Judith Hurwitz, Robin Bloor, Marcia Kaufman,FernHalper : Cloud Computing for Dummies, Wiley, 2010.	
4.	BorkoFurht, Armando Escalante (Editors) : Handbook of Cloud Computing, Springer, 2010.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Software Project Management	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST030	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This includes:

- Resolve the process of managing software from conventional to modern.
- Analyze the architecture of a model based software and the process flow.
- Describe the process automation, process management, change management, quality management, monitoring and control.

Learning Outcomes

At the end of the course, the student will be able to:

- Develop the model from the conventional software product to the modern.
- Analyze and design the software architecture.
- Have an exposure for organizing and managing a software project.
- Apply, analyze, design and develop the software project.
- Design various estimation levels of cost and effort.
- Acquire the knowledge of managing, economics for conventional, modern and future software projects.

Course Synopsis

The System Project Management (SPM) is focused on tools for planning and managing complex projects and the issues associated with complex projects. This course discusses the ways in which projects that are already underway can be monitored and tracked in terms of cost, schedule and technical progress. Various risk management techniques for identifying, tracking and mitigating risks are discussed. Further the course discusses pointers to important resources for project management, project management software tools as well as a list of empirical factors that are known to affect project success and failure.

Course Outline / Content

Unit	Topics	Week
1.	**Project Management:** The management spectrum, the people, the product, the process, the project, the W5HH principle, critical practices.	1
2.	**Metrics for Process and Project:** Metrics in the process and project Domains, software measurements, metrics for software quality, integrating metrics within software process, metrics for small organizations, establishing a software metrics program.	1
3.	**Estimation:** Observations, Project planning Process, software scope and feasibility, resources, software project estimation, decomposition techniques, empirical estimation models, estimation for object oriented projects, estimation for Agile development and web engineering projects, the make/buy decision.	2
4.	**Project Scheduling:** Basic concepts, project scheduling, defining a task set and task network, scheduling, earned value analysis. **Risk Management:** Reactive V/S proactive Risk Strategies,	2

	software risks, Risk identification, Risk projection, risk refinement, risk mitigation, monitoring and management, the RMMM plan.	
5.	**Quality Planning:** Quality Concepts, Procedural Approach to Quality Management, Quantitative Approaches to Quality Management, Quantitative Quality Management Planning, Setting the Quality Goal, Estimating Defects for Other Stages, Quality Process Planning, Defect Prevention Planning.	2
6.	**Quality Management:** Quality Concepts, Software Quality assurances, software reviews, formal technical reviews, Formalapproaches to SQA, Statistical Software Quality assurances.	1
7.	**Change Management:** Software Configuration Management, The SCM repository, SCM Process, Configuration Managementfor Web Engineering.	1
8.	**Project Execution And Closure:** Reviews. The Review Process, Planning, Overview and Preparation, Group Review Meeting, Rework and Follow-up, One-Person Review, Guidelines for Reviews in Projects, Data Collection, Analysis and Control Guidelines, Introduction of Reviews and the NAH Syndrome.	2
9.	**Project Monitoring and Control:** Project Tracking, Activities Tracking, Defect Tracking, Issues Tracking, Status Reports, Milestone Analysis, Actual Versus Estimated Analysis of Effort and Schedule, Monitoring Quality, Risk-Related Monitoring. Project Closure: Project Closure Analysis.	2
Text Books		
1.	Walker Rayce: "Software Project Management A Unified Framework", 1st Edition, Pearson Education, 2005.	
References		
1.	Richard H.Thayer: "Software Engineering Project Management", 2nd Edition, IEEE Computer Society, 1997.	
2.	Shere K.D: "Software Engineering and Management", 1st Edition, Prentice Hall, 1988.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Big Data	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST031	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course looks at concepts, technologies for big data. The focus will be on the processing techniques and algorithms that are available for a variety of "analytics". The interconnection of big data with cloud and virtualization will also be studied.

Learning Outcomes

After completion of this course, students will be able to:

- Understand how to process big data on platforms that can handle the variety, velocity, and volume of data by using a family of components that require integration and data governance.
- Familiar with the skills necessary for utilizing to handle a variety of big data analytics, and to be able to apply the analytics techniques on a variety of applications.

Course Synopsis

This course will cover important topics related to big data including Introduction of big data concepts, Examining big data types, Class Model, Study of different patterns, Case studies and memory management, Big Data NFR's, concept of distributed computing and virtualization in big data.

Course Outline / Content

Unit	Topics	Week
1.	Data Economy, Data Analytics ,Data Science, Traditional Data Processing Technologies, Large databases and their evolution. Big Data technology and trends, Big Data Introduction, Characteristics, Methodological Challenges and Problems, Example Applications.	2
2.	Examining Big Data types – Defining structured data – exploring sources and understanding role of relational databases in big data. Defining unstructured data- exploring sources and understanding role of CMS in data management.	2
3.	Providing Structure to Unstructured Data, Identification, De-identification and Reidentification. Ontologies and Semantics - Classification, Classes with Multiple Parents, Choosing a Class Model, extensible Mark-up Language.	2
4.	Introduction to Meaning, Namespaces and the Aggregation of Meaningful Assertions, Data Integration and Software Interoperability, Immutability and Immortality Application Architecture.	2
5.	Ingestion and Streaming Pattern, Storage Patterns, Access Patterns, Discovery and Analysis Patterns, Visualization Patterns, Deployment Patterns. , Case Studies. Special consideration made to the Map-Reduce paradigm. Searching, indexing, and their implications to memory management. Information extraction and feature selection. Supervised-,	2

	unsupervised-learning, and stream mining.	
6.	Big Data Computational limitations, Big Data Emerging technologies.	1
7.	Big Data NFR's., Data Privacy and Ethics, The privacy landscape, Preferences, Personalization and Relationships, Rights and Responsibility, Can data be anonymized.	1
8.	Need of distributed computing for Big Data, Virtualization and how it supports distributed computing, Cloud and Big Data, Introduction to tools used for big data.	2
Text Books		
1.	Big Data: A Revolution That Will Transform How We Live, Work, and Think by Viktor Mayer-Schönberger, Kenneth Cukier	
2.	Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends. by Michael Minelli, Michele Chambers, AmbigaDhiraj	
References		
1.	https://www.slideshare.net/nasrinhussain1/big-data-ppt-31616290	
2.	https://www.ntnu.no/iie/fag/big/lessons/lesson2.pdf	
3.	https://www.planet-data.eu/sites/default/files/presentations/Big_Data_Tutorial_part4.pdf	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Cyber Laws & Forensics	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST032	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

To maintain an appropriate level of awareness, knowledge and skill required to minimize the occurrence and severity of incidents related to forensics and cyber law.

Learning Outcomes

After completing the course you will be able to:

- Interpret and appropriately apply the laws and procedures associated with identifying, acquiring, examining and presenting digital evidence.
- Create a method for gathering, assessing and applying new and existing legislation and industry trends specific to the practice of digital forensics.
- Employ fundamental computer theory in the context of computer forensics practices.
- Adhere to the ethical standards of the profession and apply those standards to all aspects of the study and practice of digital forensics.
- Using the scientific process, apply the principles of effective digital forensics investigation techniques.

Course Synopsis

As the name suggests, Cyber Law encapsulates the legal issues related to use of the Internet. IT law covers mainly the digital information (including information security and electronic commerce) aspects and it has been described as "paper laws" for a "paperless environment". India's The Information Technology Act 2000 has tried to assimilate legal principles available in several such laws (relating to information technology) enacted earlier in several other countries, as also various guidelines pertaining to information technology law. The Act gives legal validity to electronic contracts, recognition of electronic signatures. This is a modern legislation which makes acts like hacking, data theft, spreading of virus, identity theft, defamation (sending offensive messages) pornography, child pornography, cyber terrorism, a criminal offence. It is less a distinct field of law than intellectual property or contract law, as it is a domain covering many areas of law and regulation. Some leading topics include internet access and usage, privacy, freedom of expression, and jurisdiction. Our course is specially designed to make the participant an expert of Cyber Law Fundamentals and Digital Forensics. This is made possible by discussing the in-depth concepts of computers and networks, Cyber-crime and Cyber Terrorism, the hacking techniques used by terrorist communities, encryption standards they use and other algorithms as well. Concepts of Internet Security, Digital Signature and Electronic Payment System, Digital Law, Law of Intellectual Property.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction to Forensics and Cyber Crime:** Fundamentals of computer, Internet Technology, E-Governance & E-Business ,crime, criminology, origin, source, recent trends. Emergence of information based society, economic, administration, social, dependence of use of information, accession, threats, civil society and global society, Overview of computer forensics and Investigative Techniques, Computer forensic tools, activities of forensic investigations and testing methodology.	2

2.	**Types and Categories of Cyber Crime:** Personal, Business, Financial, Office Security, Cyber Crime – Complete transparency, hacking/cracking, denial of service, IP piracy, phrasing, hetaerism etc. Cyber Attack – cyber attackers.	2
3.	**Role of Computers and Internet in Cyber crime, penetration testing and auditing :** Computer as witness, evidence, act, defining evidence, computer forensics, computer storage, media of electric record for use of course of law. Customers and legal agreements, Router penetration testing, Firewalls penetration testing, Intrusion detection system penetration testing, Wireless networks penetration testing, Password cracking penetration testing, Social engineering penetration testing, Application penetration testing, Policies and controls testing. Penetration testing report and documentation writing, Policies and procedures Security Policies-checklist.	3
4.	**Cyber Security:** The concept of cyber security , meaning, scope and the frame work, basic structure development and management, Rules, Regulations, Act, Legislation - Meaning, Scope, Difference between Rules.	2
5.	**Need for a Cyber Act:** The Indian Context , Need for a Cyber Act , Information Technology Act , Scope and further Development , Information Technology Act (Amendment) , coverage of Cyber Security and Cyber Crime Indian cyber Laws vs. cyber laws of U.S.A , similarities , scope and coverage , Effectiveness.	3
6.	**Laboratory work:** Consists of gathering information, evidence with tools like WinHex, Metasploit and Social Engineering toolkit.	2
Text Books		
1.	Cyber Forensics: from Data to Digital Evidence , Albert J. Marcella Jr., Wiley,1 st Edition,2012	
2.	Hack I.T. - Security Through Penetration Testing, T. J. Klevinsky, Scott Laliberte and Ajay Gupta, Addison-Wesley, 1st Edition,2002	
3.	Computer Forensics: Cybercriminals, Laws, And Evidence , Marie-Helen Maras, Jones & Bartlett Learn ,1st Edition ,2011.	
References		
1.	Computer Forensics: Investigating Network Intrusions and Cyber Crime, EC Council Press Series, Cengage Learning , 2010	
2.	James, S.H. and Nordby, J. J "Forensic Science – An Introduction to Scientific and Investigative Technique", CRC Press, USA (2003).	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Expert Systems	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST033	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course deals with concepts, methods, and applications of decision modeling to address various marketing issues. Unlike conventional capstone, business courses that focus on conceptual material this course will attempt to provide skills to translate conceptual understanding into developing specific operational models for improved decision-making - a skill in increasing demand in corporations today. Methodology used to transfer the knowledge of a human expert into an intelligent program that can be used to solve problems or give advice.

Learning Outcomes

After completing this course, the student should be able to:

- Apply the methodology to transfer human knowledge into an expert system.
- Apply knowledge representation.
- Design a knowledge base.
- Implement a rule-based expert system.
- Evaluate Expert System tools.
- Provide you with understanding of the role of Artificial Intelligence, Expert Systems and Decision Models in managerial decision-making.
- Develop abilities to apply, build and modify decision models to solve real problems
- Explore the issues involved in the design and development of Artificial Intelligence Based Decision Support Systems and discuss the role these systems play in the business environment.

Course Synopsis

Introduction to Expert Systems, Knowledge Representation, Inference Methods, Reasoning under Uncertainty, Inexact Reasoning, Design of Expert Systems. Machine learning and database mining.

Course Outline / Content

Unit	Topics	Week
1.	**Overview**: Background, general introduction. Forward and backward chaining, conflict resolution. Uses: structured selection, configuration, diagnosis and business rules.	2
2.	**Rule-based expert systems**: Logic and Inferences: Propositional Logic, First Order Logic, Soundness and Completeness, Forward and backward chaining. Uncertainty, fuzzy logic and belief nets. Expert System Shells.	3
3.	**Other expert system paradigms**: PIES example system (Pan and Tenenbaum) OOPs, frames, Case-based reasoning and help desks, Recommender systems (CDNow Case Study). Scheduling (Steelmaking example: Dorn and Slany).	2
4.	**Building expert systems**: CLUES example system (Talebzadeh, Mandutianu and Winner), Building expert systems Discussion of shells. Knowledge Management (Wiki web case study)	3
5.	**Machine learning and data-base mining**: AI-Agents. State	4

	Space Search: Depth First Search, Breadth First Search, DFID. Heuristic Search: Best First Search, Hill Climbing, Beam Search. Randomized Search: Simulated Annealing. Data Mining Decision Trees, Neural Networks, Text Mining, Web mining Current trends in AI.	
Text Books		
1.	The Engineering of Knowledge-based Systems, A.J. Gonzalez and D. D. Dankel, Prentice Hall, 1993.	
2.	A Guide to Expert Systems, Donald A. Waterman, Pearson publications.	
3.	Introduction to Knowledge Systems, Stefik M., Morgan Kaufmann.	
References		
1.	Giarratano J. , Riley G. , Expert Systems, Principles and Programming, PWS Publishing Company	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Mobile Computing	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST034	
Credits	3	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

To study the details of lower layers of mobile architectures in the context of pervasive computing and mobile applications.

Learning Outcomes

By the end of this course, the student will be able to:

- Understand algorithm/protocols, environments and communication systems in mobile computing;
- Have an understanding of MANETs;
- Evaluate the performance of TCP protocols in Wireless Networks with mobile nodes.

Course Synopsis

Introduction to MC; System architecture; Localization and calling; Motivation for a specialized MAC; DHCP; TCP; power aware and context-aware computing; Communications asymmetry; Wireless Application Protocol-WAP.

Course Outline / Content

Unit	Topics	Week
1.	**Mobile Computing (MC):** Introduction to MC, novel applications, limitations, and architecture.	1
2.	**GSM:** Mobile services, System architecture, Radio interface, Protocols, Localization and calling, Handover, Security, and New data services.	2
3.	**Wireless Medium Access Control:** Motivation for a specialized MAC (Hidden and exposed terminals, Near and far terminals), SDMA, FDMA, TDMA, CDMA.	2
4.	**Mobile Network Layer:** Mobile IP (Goals, assumptions, entities and terminology, IP packet delivery, agent advertisement and discovery, registration, tunnelling and encapsulation, optimizations), Dynamic Host Configuration Protocol (DHCP).	2
5.	**Mobile Transport Layer:** Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP, Fast retransmit/fast recovery, Transmission /time-out freezing, Selective retransmission, Transaction oriented TCP.	2
6.	**Database Issues:** Hoarding techniques, caching invalidation mechanisms, client server computing with adaptation, power aware and context-aware computing, transactional models, query processing, recovery, and quality of service issues.	2
7.	**Data Dissemination:** Communications asymmetry, classification of new data delivery mechanisms, pushes based mechanisms, pull-based mechanisms, hybrid mechanisms, selective tuning (indexing) techniques.	2
8.	**Mobile Ad hoc Networks (MANETs):** Wireless Application Protocol-WAP.	1

Text Books

1.	Reza B"Far, "Mobile Computing Principles and Designing and Developing Mobile Applications with UML and XML", Cambridge University Press, 2004.
2.	JochenBurkhardt, et.al." Pervasive Computing, Technology and Architecture of Mobile Internet Applications", Addison Wesley, 2002.
References	
1.	UweHansmann, LotharMerk, Martin S. Nicklous, Thomas Stober, "Principles of Mobile Computing," Springer International, 2005.
2.	Yi Bing Lin, "Wireless and Mobile Networks Architecture", John Wiley and Sons, 2000.
3.	Tomasz Imielinski et.al, "Mobile Computing", Kluwer Academic Press, 1996.
4.	UweHansmann, "Pervasive Computing Handbook. The Mobile World", IEE publication 2002.

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Green Computing	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST035	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

To acquire knowledge to adopt green computing practices to minimize negative impacts on the environment, skill in energy saving practices in their use of hardware, examine technology tools that can reduce paper waste and carbon footprint by user, and to understand how to minimize equipment disposal requirements.

Learning Outcomes

By the end of this course, the student will beable to:

- Give an account of the concept green IT;
- Give an account of environmental perspectives on IT use;
- Give an account of standards and certifications related to sustainable IT products;
- Describe green IT in relation to technology;
- Relate green IT to sustainable development;
- Evaluate IT use in relation to environmental perspectives;
- Discuss how the choice of hardware and software can facilitate a more sustainable operation;
- Use methods and tools to measure energy consumption.

Course Synopsis

Virtualization; Tele-computing; thin clients; Embedded computing and networking; Sustainable technology; Profiling Energy Usages; Green Networking; Data centre management architecture; Green Cellular Networking.

Course Outline / Content

Unit	Topics	Week
1.	**Origins**, Regulations and industry initiatives-Government, Industry. Approaches. **Virtualization**: Green maturity model for virtualization, Virtualization level: Level 0, Level 1, Level 2, Level 3.	3
2.	**Terminal servers**, Power management, Operating system support, Power supply, Storage, video card, Display. Web, temporal and spatial data mining materials recycling, Tele-computing. Thin clients: Introduction of thin clients, Characteristics of thin clients, Thin client variants.	3
3.	**Middleware support** for green computing, Tools for monitoring, HPC computing, Green Mobile, Embedded computing and networking, Management frameworks, Standards and metrics for computing green. **Environmentally Sustainable Infrastructure Design:** Sustainable technology, Sustainable intelligence, decomposing infrastructure environment. **Profiling Energy Usages for Efficient Consumption:** Profiling energy usages for the application. Profiling energy usages for the operating system and Extra energy usages profile.	4

4.	**Green Networking:** Where to save energy in wired networking, Taxonomy of green networking research: Adaptive link rate, Interface proxying, Energy ware infrastructure, Energy ware application. **Efficient-Efficient Data Canters:** Reason for over power consumption in data centers, Data center management architecture in greener perspective. **Green Cellular Networking:** Survey, Measuring greenness metrics, Energy saving in base stations, Research issues, Challenges, Future generation wireless systems, Wireless sensor network for green networking.	4
Text Books		
1.	Bud E. Smith, "Green Computing: Tools and Techniques for Saving Energy, Money, and Resources", Auerbach Publications.	
2.	Toby Velte, Anthony Velte, Robert Elsenpeter, "Green IT: Reduce Your Information System's Environmental Impact While Adding to the Bottom Line", MC-Graw Hill.	
3.	Jason Harris , "Green Computing and Green IT Best Practices on Regulations and Industry Initiatives, Virtualization, Power Management, Materials Recycling and Telecommuting", Emereo Publishing.	
References		
1.	John Lamb, "The Greening of IT-How Companies Can Make a Difference for the Environment", Pearson Education.	
2.	Greg Schulz, "The Green and Virtual Data Center", CRC Press.	
3.	F. Richard Yu, Xi Zhang, Victor C.M. Leung, "Green Communications and Networking", CRC Press.	
3.	Daniel Minoli, "Designing Green Networks and Network Operations: Saving Run-the-Engine Costs", CRC Press.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Introduction to Robotics	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST036	
Credits		**L**	**T**	**P**
Course Type	Theory			

Course Objectives

- To introduce the functional elements of Robotics.
- To impart knowledge on the direct and inverse kinematics.
- To introduce the manipulator differential motion and control.
- To educate on various path planning techniques.
- To introduce the dynamics and control of manipulators.

Course Outline / Content

Unit	Topics	Week
1.	**BASIC CONCEPTS :** Brief history-Types of Robot–Technology-Robot classifications and specifications-Design and control issues- Various manipulators – Sensors - work cell - Programming languages.	3
2.	**DIRECT AND INVERSE KINEMATICS:** Mathematical representation of Robots - Position and orientation – Homogeneous transformation-Various joints- Representation using the DenavitHattenberg parameters -Degrees of freedom-Direct kinematics-Inverse kinematics- SCARA robots-Solvability – Solution methods-Closed form solution.	3
3.	**MANIPULATOR DIFFERENTIAL MOTION AND STATICS :** Linear and angular velocities-Manipulator Jacobian-Prismatic and rotary joints–Inverse -Wrist and arm singularity - Static analysis - Force and moment Balance.	4
4.	**PATH PLANNING :** Definition-Joint space technique-Use of p-degree polynomial-Cubic polynomial-Cartesian space technique - Parametric descriptions - Straight line and circular paths - Position and orientation planning.	4
5	**DYNAMICS AND CONTROL:** Lagrangian mechanics-2DOF Manipulator-Lagrange Euler formulation-Dynamic model – Manipulator control problem-Linear control schemes-PID control scheme-Force control of robotic manipulator.	2

Text Books

1.	R.K.Mittal and I.J.Nagrath, Robotics and Control, Tata McGraw Hill, New Delhi,4th Reprint, 2005.

2.	JohnJ.Craig ,Introduction to Robotics Mechanics and Control, Third edition, Pearson Education,
3.	M.P.Groover, M.Weiss, R.N. Nageland N. G.Odrej, Industrial Robotics, McGraw-Hill Singapore, 1996.
References	
1.	AshitavaGhoshal, Robotics-Fundamental Concepts and Analysis', Oxford University Press, Sixth impression, 2010.
2.	K. K.AppuKuttan, Robotics, I K International, 2007.
3.	Edwin Wise, Applied Robotics, Cengage Learning, 2003.
4.	B.K.Ghosh, Control in Robotics and Automation: Sensor Based Integration, Allied Publishers,Chennai, 1998
5.	R.D.Klafter,T.A.Chimielewski and M.Negin, Robotic Engineering–An Integrated Approach, Prentice Hall of India, New Delhi, 1994.
6.	S.Ghoshal, " Embedded Systems & Robotics" – Projects using the 8051 Microcontroller", Cengage Learning, 2009.

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Data Analytics	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST037	
Credits		**L**	**T**	**P**
Course Type	Theory			

Course Objectives

- Be exposed to big data.
- Learn the different ways of Data Analysis.
- Be familiar with data streams.
- Learn the mining and clustering.
- Be familiar with the visualization

Course Outline / Content

Unit	Topics	Week
1.	**INTRODUCTION TO BIG DATA:** Introduction to Big Data Platform – Challenges of conventional systems - Web data – Evolution of Analytic scalability, analytic processes and tools, Analysis vs reporting - Modern data analytic tools, Stastical concepts: Sampling distributions, resampling, statistical inference, prediction error..	3
2.	**DATA ANALYSIS:** Regression modeling, Multivariate analysis, Bayesian modeling, inference and Bayesian networks, Support vector and kernel methods, Analysis of time series: linear systems analysis, nonlinear dynamics - Rule induction - Neural networks: learning and generalization, competitive learning, principal component analysis and neural networks; Fuzzy logic: extracting fuzzy models from data, fuzzy decision trees, Stochastic search methods.	3
3.	**MINING DATA STREAMS:** Introduction to Streams Concepts – Stream data model and architecture - Stream Computing, Sampling data in a stream – Filtering streams – Counting distinct elements in a stream – Estimating moments – Counting oneness in a window – Decaying window - Realtime Analytics Platform(RTAP) applications - case studies - real time sentiment analysis, stock market predictions.	4
4.	**FREQUENT ITEMSETS AND CLUSTERING:** Mining Frequent itemsets - Market based model – Apriori Algorithm – Handling large data sets in Main memory – Limited Pass algorithm – Counting frequent itemsets in a stream – Clustering Techniques – Hierarchical – K- Means – Clustering high dimensional data – CLIQUE and PROCLUS – Frequent pattern based clustering methods – Clustering in non-euclidean	4

	space – Clustering for streams and Parallelism.	
5	**FRAMEWORKS AND VISUALIZATION:** Map Reduce – Hadoop, Hive, Map R – Sharding – NoSQL Databases - S3 - Hadoopistributed file systems – Visualizations - Visual data analysis techniques, interaction techniques; Systems and applications.	2
Text Books		
1.	Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.	
2.	AnandRajaraman and Jeffrey David Ullman, Mining of Massive Datasets,Cambridge University Press, 2012.	
References		
1.	Bill Franks, Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with advanced analystics, John Wiley & sons, 2012.	
2.	Glenn J. Myatt, Making Sense of Data, John Wiley & Sons, 2007 Pete Warden, Big Data Glossary, O‟Reilly, 2011.	
3.	Jiawei Han, MichelineKamber “Data Mining Concepts and Techniques”, Second Edition, Elsevier, Reprinted 2008.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Computational Biology	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST038	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- To provide basic introduction to Systems Biology, properties of biological systems and approaches in systems biology to analyze and interpret data
- To give an overview of Synthetic Biology and analytical computational methods discussed with the help of tools and software.
- To understand the recent trends in genomics like toxico-genomics, pharmacogenomics, NGS etc.
- To familiarize the advanced topics in CADD like pharmacodynamics & pharmacokinetics.
- To introduce metabolomics with its profiling and analysis.

Learning Outcomes

By the end of this course, the student will be able to:

- Explain mathematical concepts involved in biology
- Gain basic knowledge of modern molecular biology and genomics
- Develop an algorithm for analysis of biological sequences.
- Gain knowledge to identify and develop in silico models appropriate to the different biological projects.
- Apply molecular methods to study genetic variation within and between species 6) Explain and evaluate different phylogenetic optimal criteria.
- Correctly select systems biology tools that will help them in re-constructing and redefining complex biological processes.

Course Synopsis

System Biology, Gene Regulatory Network, Protein Interaction Network,Sysnthetic Biology, Computational Synthetic biology,Engineering Biology, Toxic-genomics, Pharmacogenomics, Pharmaco-genetics, Molecular dynamics simulations, Metabolome informatics.

Course Outline / Content

Unit	Topics	Week
1.	**Systems biology:** Self-organization, emergence, modularity and abstraction, feedback, control analysis, Enzyme Kinetics and **Thermodynamics:** The Law of Mass Action; Reaction Kinetics, Rate Equation, Michaelis-Menten Equation, Hill Equation, Interaction networks overview- Gene Regulatory Network, Protein – Protein Interaction Network, Signaling Pathways, Metabolic path-ways; network motifs, **Systems Biology tools and standards:** Matlab Systems Biology toolbox; SBML; SBGL (Systems Biology Graphical Language); KEGG; Tools for systems Biology- Cell designer; Cytoscape.	3

2.	**Synthetic Biology:** Engineering Biology; design and construction of novel biological systems; Abstraction hierarchy-Part, Device, Systems; Bio-Bricks - a standard for (physical) DNA composition, Designing a biological system from Bio-bricks; iGEM; SBOL, Computational Synthetic biology: Codon optimization; AND gate and OR gate in biology; Operons; Switches and clocks; Re-pressilator; Applications- Environment, Energy, Pharmaceutical needs, Ethical issues of Synthetic Biology.	3
3.	**Niche areas in Genomics:** Toxic-genomics, Pharmacogenomics, Pharmaco-genetics, SNP, Personalized medicine, Meta-genomics, Comparative genomics, Functional genomics, structural genomics, QTL, HGP. Next Generation Sequencing methods, Overview of data compression, Need for compression, Scope of NGS data compression.	3
4.	**Advanced topics in CADD:** Molecular dynamics simulations, Force fields, Energy minimization, pharmacodynamics & pharmacokinetics, 2D and 3D screening, Identification of targets in silico, GPCRs, Peptides as drugs, introduction to Ayur-informatics.	3
5.	**Metabolomics:** Metabolism, metabolomite, metabolome, metabolomic separation and analysis techniques, metabolic profiling, metabolic fingerprinting, Metabolome informatics. Resources/databases of metabolomics, Applications; Epigenetics	3
Text Books		
1.	Alon, U. (2006). An introduction to systems biology: design principles of biological circuits. CRC press.	
2.	Gautham, N. (2006). Bioinformatics: Databases and Algorithms. Alpha Science Int'l Ltd.	
3.	Benson, G. (2003). Algorithms in Bioinformatics. Springer Berlin Heidelber	
References		
1.	Choi, S. (Ed.). (2007). Introduction to systems biology. New Jersey:: Humana Press.	
2.	Demin, O., &Goryanin, I. (2010). Kinetic modelling in systems biology. CRC Press.	
3.	Gusfield, D. (1997). Algorithms on strings, trees and sequences: computer science and computational biology. Cambridge University Press.	
4.	Iyengar, S. (2010). Symbolic Systems Biology. Jones and Bartlett.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Special Topics in Computer Science	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST039	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course will introduce students to this rapidly growing field and equip them with some of its basic principles and tools as well as its general mindset. Students will learn concepts, techniques and tools they need to deal with various facets of data science practice, including data collection and integration, exploratory data analysis, predictive modeling, descriptive modeling, data product creation, evaluation, and effective communication.

Learning Outcomes

After completion of this course students will be able to:

- Explain what Data Science is and the skill sets needed to be a data scientist.
- Apply basic machine learning algorithms (Linear Regression, k-Nearest Neighbors (k-NN), k-means, Naive Bayes) for predictive modeling.
- Identify probability distributions commonly used as foundations for statistical modeling. Fit a model to data.
- Explain the significance of exploratory data analysis (EDA) in data science. Apply basic tools (plots, graphs, summary statistics) to carry out EDA.
- Build their own recommendation system using existing components.
- Reason around ethical and privacy issues in data science

Course Synopsis

Data Science is the study of the generalizable extraction of knowledge from data. Being a data scientist requires an integrated skill set spanning mathematics, statistics, machine learning, databases and other branches of computer science along with a good understanding of the craft of problem formulation to engineer effective solutions.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction:** Introduction to Data Science, Steps in doing data Science, Skills needed to do data Science, Datafication	1
2.	**Statistical Inference** Populations and samples, Statistical modelling, probability distributions, fitting a model, Intro to R	1
3.	**Exploratory Data Analysis and the Data Science Process** - Basic tools (plots, graphs and summary statistics) of EDA, Philosophy of EDA, The Data Science Process, Case Study: RealDirect (online real estate firm)	2
4.	**Introduction to Machine Learning**: Linear Regression ,K-Nearest Neighbors (K-NN), k-means, Motivating application: Filtering Spam , Why Linear Regression and k-NN are poor choices for Filtering Spam , Naive Bayes and why it works for	2

	Filtering Spam , Data Wrangling: APIs and other tools for scrapping the Web	
5.	**Feature Generation and Feature Selection (Extracting Meaning From Data) -** Motivating application: user (customer) retention Feature Generation (brainstorming, role of domain expertise, and place for imagination) ,Feature Selection algorithms – Filters; Wrappers; Decision Trees; Random Forests.	2
6.	**Recommendation Systems**: Building a User-Facing Data Product, Algorithmic ingredients of a Recommendation Engine, Dimensionality Reduction, Singular Value Decomposition, Principal Component Analysis, Exercise: build your own recommendation system	2
7.	**Mining Social-Network Graphs:** Social networks as graphs, Clustering of graphs, Direct discovery of communities in graphs, Partitioning of graphs, Neighborhood properties in graphs.	1
8.	**Data Visualization** Basic principles, ideas and tools for data visualization, Examples of inspiring (industry) projects, Exercise: create your own visualization of a complex dataset.	1
9.	**Data Science and Ethical Issues**: Discussions on privacy, security, ethics, A look back at Data Science , Next-generation data scientists	1

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	System & Network Administration	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST040	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

5. Understand the role and responsibilities of a system administrator
6. Configure the Unix operating system
7. Focuses on the principles and techniques used in the design of networks and development of networked and distributed software.
8. Students will be exposed to standard network design tools, and diagnostic tools, such as packet monitors and performance analysis tools.

Learning Outcomes

- Demonstrate an understanding of system components, the advantages of Unix OS.
- Design and diagnosis of networks will also be covered.
- Systems Administration will also be discussed, especially with respect to Network Management; However, homework and programming assignments will be more focused on Network programming.

Course Synopsis

The maintenance and deployment of computer systems in production environments requires significant effort. This course distils decades of experience into operational principles that apply across technologies.

Course Outline / Content

Unit	Topics	Week
1.	Introduction to Networks, OSI interconnect model, topologies, Internet history and TCP/IP	1
2.	Physical Layer: transmission media, socket programming, UNIX Process Creation and UNIX IPC.	1
3.	Introduction to System administration as a discipline: its goals, philosophy, challenges and common practices, Discussion of computer system components and Operating systems components: Unix-like systems vs Windows systems	1
4.	Data Link Layer: framing, flow control, error control, encoding for local and wide areas, Admin tricks with UNIX shell.	2
5	Medium Access Layer. Broadcast, CSMA/CD, CDMA, FDDI, 802.X, Bluetooth.	1
6	Network Layer: Flow control, congestion control, Routing,. quality of service, switching, CIDRs, mobile IP, WAP	2
7	Finish Routing. Transport Layer: TCP, UDP, IP v 6. CISCO Router IOS.	1
8	Application Layer: httpd, smtp, dns, snmp, ftp, Telnet, streaming video, video compression, multicast, JME. Network Services. Dist Computing, Network Management	2

Text Books

1.	Nemeth, Snyder, Hein and Whaley "UNIX and Linux System Administration Handbook", 4th Ed.(Prentice Hall, 2010)
2.	William Stallings: Data & Computer Communications, 7th Ed,PHI
References	
4.	AndrewTanenbaum,–Computer Networks‖PHI

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Pattern Recognition	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST041	
Credits		**L**	**T**	**P**
Course Type	Theory			

Course Objectives

- Understand the concept of a pattern and the basic approach to the development of pattern recognition and machine intelligence algorithms.
- Understand the basic methods of feature extraction, feature evaluation, and data mining.
- Understand and apply both supervised and unsupervised classification methods to detect and characterize patterns in real-world data.
- Develop prototype pattern recognition algorithms that can be used to study algorithm behavior and performance against real-world multivariate data.

Learning Outcomes

After the course the student should be able to

- Define basic concepts in modelling and simulation
- Classify various simulation models and give practical examples for each category
- Construct a model for a given set of data and motivate its validity
- Analyze output data produced by a model and test validity of the model

Course Synopsis

Course Outline / Content

Unit	Topics	Week
1.	Introduction to patterns and pattern recognition application development: Supervised pattern detection I (Bayes classifiers) Feature extraction - multivariate data Feature extraction - image data Supervised pattern detection II (linear classifiers) Unsupervised pattern detection I (clustering) Supervised pattern detection III (non-linear classifiers, neural networks, support vector machines) Supervised pattern detection IV (rule-based classifiers) Unsupervised pattern detection II (self-organization, competitive learning)	2
2.	**Basics of Probability, Random Processes and Linear Algebra (recap)**: Probability: independence of events, conditional and joint probability, Bayes theorem Random Processes: Stationary and non-stationary processes, Expectation, Autocorrelation, Cross-Correlation, spectra.	2
3.	**Bayes Decision Theory**: Minimum-error-rate classification. Classifiers, Discriminant functions, Decision surfaces. Normal density and discriminant functions. Discrete features.	2

4.	**Parameter Estimation Methods**: Maximum-Likelihood estimation: Gaussian case. Maximum a Posteriori estimation. Bayesian estimation: Gaussian case. Unsupervised learning and clustering - Criterion functions for clustering. Algorithms for clustering: K-Means, Hierarchical and other methods. Cluster validation. Gaussian mixture models, Expectation-Maximization method for parameter estimation. Maximum entropy estimation. Sequential Pattern Recognition. Hidden Markov Models (HMMs). Discrete HMMs. Continuous HMMs. Nonparametric techniques for density estimation. Parzen-window method. K-Nearest Neighbor method	3
5.	**Dimensionality reduction**: Principal component analysis - it relationship to eigen analysis. Fisher discriminant analysis - Generalized eigen analysis. Eigen vectors/Singular vectors as dictionaries. Factor Analysis, Total variability space - a dictionary learning methods. Non-negative matrix factorization - a dictionary learning method.	2
6.	**Linear discriminant functions**: Gradient descent procedures, Perceptron, Support vector machines - a brief introduction.	1
7.	**Artificial neural networks**: Multilayer perceptron - feedforward neural network. A brief introduction to deep neural networks, convolutional neural networks, recurrent neural networks.	2
8.	**Non-metric methods for pattern classification**: Non-numeric data or nominal data. Decision trees: Classification and Regression Trees (CART).	1

Text Books

1.	R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley
2	S.Theodoridis and K.Koutroumbas, Pattern Recognition, 4th Ed.
3	C.M.Bishop, Pattern Recognition and Machine Learning, Springer
4	MATLAB latest release full.

Student expectations

This course will aim at basic student requirements:

Case studies - A number of case studies will be assigned to students throughout the term and will provide an opportunity for the student to work with real world data to build realistic pattern recognition applications. Students are also required to select and develop a case study of their own choosing by the end of the semester, and to present their case studies to the rest of the class during the last class of the semester.

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Natural Language Processing	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST042	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- This course is about a variety of ways to represent human languages (like English and Chinese) as computational systems, and how to exploit those representations to write programs that do useful things with text and speech data, like translation, summarization, extracting information, question answering, natural interfaces to databases, and conversational agents.
- This field is called Natural Language Processing or Computational Linguistics, and it is extremely multidisciplinary. This course will therefore include some ideas central to Machine Learning (discrete classification, probability models) and to Linguistics (morphology, syntax, semantics).

Learning Outcomes

- We'll cover computational treatments of words, sounds, sentences, meanings, and conversations. We'll see how probabilities and real-world text data can help. We'll see how different levels interact in state-of-the-art approaches to applications like translation and information extraction

Course Synopsis

The course is designed for SCS undergraduate students, and also to students in graduate programs who have a peripheral interest in natural language, or linguistics students who know how to pro1 gram. Prerequisite: Fundamental Data Structures and Algorithms (15-211) or equivalent; strong programming capabilities.

Course Outline / Content

Unit	Topics	Week
1.	Introduction- Human languages, models, ambiguity, processing paradigms; Phases in natural language processing, applications. Text representation in computers, encoding schemes. Linguistics resources- Introduction to corpus, elements in balanced corpus, TreeBank, PropBank, WordNet, VerbNet etc. Resource management with XML, Management of linguistic data with the help of GATE, NLTK.	2
2.	Regular expressions, Finite State Automata, word recognition, lexicon. Morphology, acquisition models, Finite State Transducer. N-grams, smoothing, entropy, HMM, ME, SVM, CRF. Part of Speech tagging- Stochastic POS tagging, HMM, Transformation based tagging (TBL), Handling of unknown words, named entities, multi word expressions.	3
3.	A survey on natural language grammars, lexeme, phonemes, phrases and idioms, word order, agreement, tense, aspect and mood and agreement, Context Free Grammar, spoken language syntax	4
4.	Parsing- Unification, probabilistic parsing, TreeBank. Semantics- Meaning representation, semantic analysis, lexical semantics, WordNet Word Sense Disambiguation- Selectional restriction,	3

	machine learning approaches, dictionary based approaches. Discourse- Reference resolution, constraints on co-reference, algorithm for pronoun resolution, text coherence, discourse structure	
5.	Applications of NLP- Spell-checking, Summarization Information Retrieval- Vector space model, term weighting, homonymy, polysemy, synonymy, improving user queries. Machine Translation– Overview.	3
Text Books		
1.	Daniel Jurafsky and James H Martin. Speech and Language Processing, 2e, Pearson Education, 2009	
2.	James A.. Natural language Understanding 2e, Pearson Education, 1994	
References		
1.	Bharati A., Sangal R., Chaitanya V.. Natural language processing: a Paninian perspective, PHI, 2000	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Quantum Computing	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST043	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course is designed:

- To enable students with non-physics backgrounds to 'think quantumly'
- To recognize which classical assumptions fall apart at the quantum level
- To begin to reintegrate the strange results of quantum theory into the broader framework of classical computer science

Learning Outcomes

Enable the student to:

- Translate fluently between the major mathematical representations of quantum operations.
- Implement basic quantum algorithms.
- To acquire a working knowledge of quantum information theory.

Course Synopsis

This course is an introduction to quantum information theory (qubits, quantum gates, and qubit systems). It covers a few selected quantum algorithms, yet the emphasis of the course is on quantum simulation (i.e. quantum informational representations of quantum systems and quantum algorithms for computational physics).

Course Outline / Content

Unit	Topics	Week
1.	**Introduction and Background:** Overview, Computers and the Strong Church–Turing Thesis, The Circuit Model of Computation, A Linear Algebra Formulation of the Circuit Model, Reversible Computation, A Preview of Quantum Physics, Quantum Physics and Computation. **Linear Algebra and The Dirac Notation**: The Dirac Notation and Hilbert Spaces, Dual Vectors, Operators, The Spectral Theorem, Functions of Operators, Tensor Products, The Schmidt Decomposition Theorem, Some Comments on the Dirac Notation.	2
2.	**Qubits and The Framework of Quantum Mechanics:** The State of a Quantum System, Time-Evolution of a Closed System, Composite Systems, Measurement, Mixed States and General Quantum Operations. **A Quantum Model of Computation:** The Quantum Circuit Model, Quantum Gates, Universal Sets of Quantum Gates, Efficiency of Approximating Unitary Transformations, Implementing Measurements with Quantum Circuits.	2
3.	**Superdense Coding and Quantum Teleportation:** Superdense Coding, Quantum Teleportation, An Application of Quantum Teleportation. **Introductory Quantum Algorithms:** Probabilistic Versus Quantum Algorithms, Phase Kick-Back, The Deutsch	2

	Algorithm, The Deutsch–Jozsa Algorithm, Simon's Algorithm.	
4.	**Algorithms With Superpolynomial Speed-Up:** Quantum Phase Estimation and the Quantum Fourier Transform, Eigenvalue Estimation, Finding-Orders, Finding Discrete Logarithms, Hidden Subgroups, Related Algorithms and Techniques. **Algorithms Based on Amplitude Amplification:** Grover's Quantum Search Algorithm, Amplitude Amplification, Quantum Amplitude Estimation and Quantum Counting, Searching Without Knowing the Success Probability, Related Algorithms and Techniques	4
5.	**Quantum Computational Complexity Theory and Lower Bounds:** Computational Complexity, The Black-Box Model, Lower Bounds for Searching in the Black-Box Model: Hybrid Method, General Black-Box Lower Bounds, Polynomial Method, Block Sensitivity, Adversary Methods. **Quantum Error Correction:** Classical Error Correction, The Classical Three-Bit Code, Fault Tolerance, Quantum Error Correction, Three- and Nine-Qubit Quantum Codes, Fault-Tolerant Quantum Computation.	4
Text Books		
1.	Eleanor G. Rieffel and Wolfgang H. Polak, "Quantum Computing: A Gentle Introduction"	
References		
1.	Michael A. Nielsen and Isaac L. Chuang, "Quantum Computation and Quantum Information".	
2.	Phillip Kaye, Raymond Laflamme, and Michele Mosca (2007). An Introduction to Quantum Computing. Oxford University Press.	
3.	Yanofsky, Noson S. and Mirco A. Mannucci (2008). Quantum Computing for Computer Scientists. Cambridge University Press.	
4.	McMahon, David (2008). Quantum Computing Explained. John Wiley & Sons, Inc.	
5.	Mermin, N. David (2007). Quantum Computer Science: An Introduction. Cambridge University Press.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Deep Learning	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST044	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- This course is an introduction to deep learning, a branch of machine learning concerned with the development and application of modern neural networks. Deep learning algorithms extract layered high-level representations of data in a way that maximizes performance on a given task. For example, asked to recognize faces, a deep neural network may learn to represent image pixels first with edges, followed by larger shapes, then parts of the face like eyes and ears, and, finally, individual face identities. Deep learning is behind many recent advances in AI.

Learning Outcomes

- This is an upper-level undergraduate/graduate course.
- All students should have the following skills:
- Calculus, Linear Algebra ■ Probability & Statistics ■ Ability to code in Python

Course Synopsis

We will cover a range of topics from basic neural networks, convolutional and recurrent network structures, deep unsupervised and reinforcement learning, and applications to problem domains like speech recognition and computer vision. Prerequisites: a strong mathematical background in calculus, linear algebra, and probability & statistics (students will be required to pass a math prerequisites test), as well as programming in Python and C/C++. There will be assignments and a final project.

Course Outline / Content

Unit	Topics	Week
1.	Feedforward Neural networks. Gradient descent and the back propagation algorithm. Unit saturation, aka the vanishing gradient problem, and ways to mitigate it. RelU Heuristics for avoiding bad local minima. Heuristics for faster training. Nestors accelerated gradient descent. Regularization. Dropout.	2
2.	Convolutional Neural Networks: Architectures, convolution / pooling layers Recurrent Neural Networks: LSTM, GRU, Encoder Decoder architectures	3
3.	Deep Unsupervised Learning: Autoencoders (standard, sparse, denoising, contractive, etc), Variational Autoencoders, Adversarial Generative Networks, Autoencoder and DBM Attention and memory models, Dynamic memory networks. Applications of Deep Learning to Computer Vision: Image segmentation, object detection, automatic image captioning, Image generation with Generative adversarial networks, video to text with LSTM models. Attention models for computer vision tasks.	4
4.	Applications of Deep Learning to NLP: Introduction to NLP and Vector Space Model of Semantics: Word Vector Representations: Continuous Skip-Gram Model, Continuous	3

	Bag-of Words model (CBOW), Glove, Evaluations and Applications in word similarity, analogy reasoning: Named Entity Recognition, Opinion Mining using Recurrent Neural Networks	
5.	Named Entity Recognition, Opinion Mining using Recurrent Neural Networks: Parsing and Sentiment Analysis using Recursive Neural Networks: Sentence Classification using Convolutional Neural Networks: Dialogue Generation with LSTMs (1 lecture) Applications of Dynamic Memory Networks in NLP (1 lecture) Recent Reseearch in NLP using Deep Learning: Factoid Question Asnwering, similar question detection, Dialogue topic tracking, Neural Summarization, Smart Reply	3
Text Books		
1.	Bengio, Yoshua, Ian J. Goodfellow, and Aaron Courville. "Deep learning." An MIT Press book in preparation. (2015)	
2.	Bengio, Yoshua. "Learning deep architectures for AI." Foundations and trends in Machine Learning 2.1 (2009): 1127.	
References		
1.	Hochreiter, Sepp, and Jargen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 17351780.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Introduction to Data Science	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST045	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course will introduce students to this rapidly growing field and equip them with some of its basic principles and tools as well as its general mindset. Students will learn concepts, techniques and tools they need to deal with various facets of data science practice, including data collection and integration, exploratory data analysis, predictive modeling, descriptive modeling, data product creation, evaluation, and effective communication.

Learning Outcomes

After completion of this course students will be able to:

- Explain what Data Science is and the skill sets needed to be a data scientist.
- Apply basic machine learning algorithms (Linear Regression, k-Nearest Neighbors (k-NN), k-means, Naive Bayes) for predictive modeling
- Identify probability distributions commonly used as foundations for statistical modeling. Fit a model to data.
- Explain the significance of exploratory data analysis (EDA) in data science. Apply basic tools (plots, graphs, summary statistics) to carry out EDA.
- Build their own recommendation system using existing components.
- Reason around ethical and privacy issues in data science

Course Synopsis

Data Science is the study of the generalizable extraction of knowledge from data. Being a data scientist requires an integrated skill set spanning mathematics, statistics, machine learning, databases and other branches of computer science along with a good understanding of the craft of problem formulation to engineer effective solutions

Course Outline / Content

Unit	Topics	Week
1.	**Introduction:** Introduction to Data Science, Steps in doing data Science, Skills needed to do data Science, Datafication	1
2.	**Statistical Inference** Populations and samples, Statistical modelling, probability distributions, fitting a model, Intro to R	1
3.	**Exploratory Data Analysis and the Data Science Process** - Basic tools (plots, graphs and summary statistics) of EDA, Philosophy of EDA, The Data Science Process, Case Study: RealDirect (online real estate firm)	2
4.	**Introduction to Machine Learning**: Linear Regression ,K-Nearest Neighbors (K-NN), k-means, Motivating application: Filtering Spam , Why Linear Regression and k-NN are poor choices for Filtering Spam , Naive Bayes and why it works for Filtering Spam , Data Wrangling: APIs and other tools for scrapping the Web	2
5.	**Feature Generation and Feature Selection (Extracting Meaning From Data)** - Motivating application: user (customer)	2

	retention Feature Generation (brainstorming, role of domain expertise, and place for imagination) ,Feature Selection algorithms – Filters; Wrappers; Decision Trees; Random Forests	
6.	**Recommendation Systems**: Building a User-Facing Data Product, Algorithmic ingredients of a Recommendation Engine, Dimensionality Reduction, Singular Value Decomposition, Principal Component Analysis, Exercise: build your own recommendation system	2
7.	**Mining Social-Network Graphs:** Social networks as graphs, Clustering of graphs, Direct discovery of communities in graphs, Partitioning of graphs, Neighborhood properties in graphs.	1
8.	**Data Visualization** Basic principles, ideas and tools for data visualization, Examples of inspiring (industry) projects, Exercise: create your own visualization of a complex dataset.	1
9.	**Data Science and Ethical Issues**: Discussions on privacy, security, ethics, A look back at Data Science , Next-generation data scientists	1
Text Books		
1.	Cathy O'Neil and Rachel Schutt. Doing Data Science, Straight Talk From The Frontline. O'Reilly. 2014.	
2.	Jure Leskovek, Anand Rajaraman and Jeffrey Ullman. Mining of Massive Datasets. v2.1, Cambridge University Press. 2014.	
3.	Foster Provost and Tom Fawcett. Data Science for Business: What You Need to Know about Data Mining and Data-analytic Thinking. ISBN 1449361323. 2013.	
References		
1.	Foster Provost and Tom Fawcett. Data Science for Business: What You Need to Know about Data Mining and Data-analytic Thinking. ISBN 1449361323. 2013.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Internet Of Things	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST046	
Credits	03	**L**	**T**	**P**
Course Type	Theory			

Course Objectives

This course will enable students to

- •Define and explain basic issues, policy and challenges in the IOT
- •Illustrate Mechanism and Key Technologies in IOT
- •Explain the Standard of the IOT
- •Explain resources in the IOT and deploy of resources into business
- •Demonstrate data analytics for IOT

Learning Outcomes

At the end of this course the students will be able to:

- •Develop schemes for the applications of IOT in real time scenarios
- •Manage the Internet resources
- •Model the Internet of things to business
- •Understand the practical knowledge through different case studies
- •Understand data sets received through IOT devices and tools used for analysis

Course Synopsis

The course deals with all the important aspects of discrete Internet of things and to develop schemes for the applications of IOT in real time scenarios This course is meant for an upper level undergraduate or master's level introduction to manage the internet resources and gain the practical knowledge through different case studies.

Course Outline / Content

Unit	Topics	Week
1.	What is The Internet of Things? Overview and Motivations, Examples of Applications, IPV6 Role, Areas of Development and Standardization, Scope of the Present Investigation. Internet of Things Definitions and frameworks-IOT Definitions, IOT Frameworks, Basic Nodal Capabilities. Internet of Things Application Examples-Overview, Smart Metering/Advanced Metering Infrastructure-Health/Body Area Networks, City Automation, Automotive Applications, Home Automation, Smart Cards, Tracking, Over-The-Air-Passive Surveillance/Ring of Steel, Control Application Examples, Myriad Other Applications.	2
2.	Fundamental IOT Mechanism and Key Technologies-Identification of IOT Object and Services, Structural Aspects of the IOT, Key IOT Technologies. Evolving IOT Standards-Overview and Approaches, IETF IPV6 Routing Protocol for RPL Roll, Constrained Application Protocol, Representational State Transfer, ETSI M2M,Third Generation Partnership Project Service Requirements for Machine-Type Communications, CENELEC, IETF IPv6 Over Low power WPAN, Zigbee IP(ZIP),IPSO	3
3	Layer ½ Connectivity: Wireless Technologies for the IOT-	3

	WPAN Technologies for IOT/M2M, Cellular and Mobile Network Technologies for IOT/M2M, Layer 3 Connectivity: IPv6 Technologies for the IOT: Overview and Motivations. Address Capabilities, IPv6 Protocol Overview, IPv6 Tunnelling, IPSec in IPv6, Header Compression Schemes, Quality of Service in IPv6, Migration Strategies to IPv6.	
4.	Case Studies illustrating IOT Design-Introduction, Home Automation, Cities, Environment, Agriculture, Productivity Applications.	2
5.	Data Analytics for IOT –Introduction, Apache Hadoop, Using Hadoop Map Reduce for Batch Data Analysis, Apache Oozie, Apache Spark, Apache Storm, Using Apache Storm for Real-time Data Analysis, Structural Health Monitoring Case Study.	2
Text Books		
1.	Daniel Minoli,"Building the Internet of Things with IPv6 and MIPv6:The Evolving World of M2M Communications", Wiley, 2013.	
2.	Arshdeep Bahga, Vijay Madisetti, "Internet of Things: A Hands on Approach" Universities Press., 2015	
References		
1.	Michael Miller," The Internet of Things", First Edition, Pearson, 2015.	
2.	Claire Rowland,Elizabeth Goodman et.al.," Designing Connected Products", First Edition,O'Reilly, 2015.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Advanced Cryptography	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST047	
Credits	3	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- To learn fundamental concepts of computer security and cryptography and utilize these techniques in computing systems.
- To learn about Pseudo-random Generators (PRG), building a Pseudorandom Permutation and its applications.
- To develop an understanding of Message Authentication Codes (MACs) and Public Key Signature Schemes.

Learning Outcomes

- The objectives of this course are to:
- 1. To understand the fundamentals of Cryptography
- 2. To acquire knowledge on standard algorithms used to provide confidentiality, integrity and authenticity.
- 3. To understand the various key distribution and management schemes.
- 4. To understand how to deploy encryption techniques to secure data in transit across data networks
- 5. To design security applications in the field of Information technology

Course Synopsis

Fundamentals of Cryptography; Basics of Symmetric and Asymmetric Key Cryptography; Notions of semantic Security and various types of Attacks; Weak and Strong one way functions; Building a Pseudorandom Permutation; Message Authentication Codes (MACs);Various Public Key Signature Schemes.

Course Outline / Content

Unit	Topics	Week
1.	**Introduction** :Attacks on computers and computer security, need for security, approaches , principles, types of attacks ,operational model of network security Cryptography concepts and techniques, substitution, transposition, encryption and decryption, symmetric, Asymmetric key cryptography, key range size, possible type of attacks.	1
2.	Mathematics of cryptography and DES Block ciphers modes, feistel ciphers DES. working of DES ,cracking des ,problems on des., 2DES, 3DES, des design ,Side channel attacks, Differential cryptanalysis.	2
3.	Symmetric-Key Cryptography: Glosis field theory, AES, overview of Rijndael - comparison with others. Symmetric ciphers, Blowfish in practice ,RC4, RC5,RC6,IDEA, RSA	3
4.	Asymmetric-Key Cryptography RSA, Elliptic curve cryptography ECC, Digital certificates and PK	4
5.	Cryptographic Hash Functions Hashing schemes SHA-family, MAC, Digital Signature RSA El Gomel , DSS DSA, Authentication Protocols , applications Kerberos, X.509 Directory services	5

6.	Network Security Internet security protocols , SSL,TLS TSP WAP security, SET Hashing Authentication & Signature Schemes E-mail security, Email architecture SSL, PGP, MIME, S/MIME Internet Protocol Security (IPSec) IPSec architecture, IPSec verses other layers security Mobile IPSec, VPN, Web security SSL, TLS, SET etc	6
7	System Security Intruders, types of attacks, protecting against Intruders honeypots, scanning and analysis tools, Viruses and worms, types of viruses, protection, Firewall architecture implementing firewalls, xml firewalls, trusted systems, trusted system applications, multi-level security, trusted products. Security implementation, wireless security, securities in Adhoc-networks.	7
Text Books		
1.	Cryptography And Network Security Principles and Practices William Stallings, Prentice Hall	
2.	Cryptography and Network Security Atul Kahate, Tata McGraw-Hill	
3.	Cryptography and Network Security Behrouz A. Forouzan, TMH	
4.	Wade Trappe, Lawrence C Washington, " Introduction to Cryptography with coding theory", Pearson.	
References		
1.	W. Mao, "Modern Cryptography – Theory and Practice", Pearson Education	
2.	Charles P. Pfleeger, Shari Lawrence Pfleeger – Security in computing – Prentice Hall of India.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Data Mining	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST048	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- To introduce students to the basic concepts and techniques of Data Mining.
- To develop skills of using recent data mining software for solving practical problems.
- To gain experience of doing independent study and research.
- To study the methodology of engineering legacy databases for data warehousing and data mining to derive business rules for decision support systems.

Learning Outcomes

This course has the following program learning outcomes:

- Study the major data mining problems as different types of computational tasks (prediction, classification, clustering, etc.) and the algorithms appropriate for addressing these tasks.
- Learn how to analyze data through statistical and graphical summarization, supervised and unsupervised learning algorithms.
- Systematically evaluate data mining algorithms and understand how to choose algorithms for different analysis tasks.

Course Synopsis

This course introduces basic concepts, tasks, methods, and techniques in data mining. The emphasis is on various data mining problems and their solutions. Students will develop an understanding of the data mining process and issues, learn various techniques for data mining, and apply the techniques in solving data mining problems using data mining tools and systems. Students will also be exposed to a sample of data mining applications.

Course Outline / Content

Unit	Topics	Week
2.	Introduction to Data Mining: Introduction: Scope of Data Mining: What is Data Mining; How does Data Mining Works, Predictive Modelling: Data Mining and Data Warehousing: Architecture for Data Mining: Profitable Applications: Data Mining Tools	1
3.	Business Intelligence: Introduction, Business Intelligence, Business Intelligence tools, Business Intelligence Infrastructure, Business Intelligence Applications, BI versus Data Warehouse, BI versus Data Mining, Future of BI.	2
4.	Data Pre-processing: Introduction, Data Pre-processing Overview, Data Cleaning, Data Integration and Transformation, Data Reduction, Discretization and Concept Hierarchy Generation.	2
5.	Data Mining Techniques- An Overview: Introduction, Data Mining, Data Mining Versus Database Management System, Data Mining Techniques- Association rules, Classification, Regression, Clustering, Neural networks.	2

6.	Clustering: Introduction, Clustering, Cluster Analysis, Clustering Methods- K means, Hierarchical clustering, Agglomerative clustering, Divisive clustering, clustering and segmentation software, evaluating clusters.	1
7.	Web Mining: Introduction, Terminologies, Categories of Web Mining – Web Content Mining, Web Structure Mining, Web Usage Mining, Applications of Web Mining, and Agent based and Data base approaches, Web mining Software.	2
8.	Applications of Data mining: Introduction, Business Applications Using Data Mining- Risk management and targeted marketing, Customer profiles and feature construction, Medical applications (diabetic screening), Scientific Applications using Data Mining, Other Applications.	2
Text Books		
3.	Kamber and Han, "Data Mining Concepts and Techniques", Hart Court India P. Ltd. Elsevier Publications Second Edition, 2001	
4.	Paul Raj Poonia, "Fundamentals of Data Warehousing", John Wiley & Sons, 2004.	
References		
5.	W. H. Inmon, "Building the operational data store", 2nd Ed., John Wiley, 1999.	
6.	Pang- Ning Tan, Michael Steinbach, Viach, Vipin Kumar, Introduction to Data Mining, Pearson	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Advanced Graph Algorithms	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST049	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

- To explain the major graph algorithms and their analyses.
- To identify different parameters of graphs and its applications.
- To understand planar graphs and its properties of a given properties to detect planarity of a given graph.
- To apply optimization techniques to construct a minimal spanning tree of a graph, prefix code for a given message.

Learning Outcomes

Students who complete the course will have demonstrated the ability to do the following:

- Argue the correctness of algorithms using inductive proofs and invariants.
- Analyze worst-case running times of algorithms using asymptotic analysis.
- Employ graphs to model engineering problems, when appropriate.
- Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them.
- Compare between different data structures. Pick an appropriate data structure for a design situation.

Course Synopsis

This course covers advanced graph algorithms from various fields.. Fast algorithms for fundamental graph optimization problems, including maximum flow, minimum cuts, minimum spanning trees, nonbipartite matching, planar separators and applications, and shortest paths.

Course Outline / Content

Unit	Topics	Week
1.	**Basics:** Introduction, Machine Model, Graph Data Structures, Bipartite Graphs, Eulerian Graphs, Circuits & Trails, Fleury's Algorithm, Hierholzer's Algorithm	2
2.	**Connectivity:** Top. Sort, Detecting Strong Components, 2-Connectivity / 2-Edge-Connectivity, (Open) Ear Decompositions, Strong Orientations, Testing 2-(Edge-)Connectivity in Linear Time, Bipolar Orientations, s-t-Numberings + Algorithm	3
3.	**Matchings:** Definitions, Hopcroft–Karp algorithm, Edmonds algorithm, Hall's Theorem, Hungarian Method for Bipartite Weighted Matchings, Weighted Matchings in General Graphs, Some approximate approach	2
4.	**Dynamic Algorithms:** Dynamic Connectivity and Spanning Trees in Amortized Poly-log time, Dynamic Connectivity in Worst-case O(n½) time	2

5.	**Planar Graphs:** Planar Separator Theorem and its Applications, Embeddings (combinatorial + planar), Euler's Formula, Kuratowski's Theorem, Detour to Platonic Solids, Dual Graphs, Interdigitating Trees, Half-Edge Data Structure, Decremental Dynamic Adjacency Queries, Max-Cut in polynomial time, Minimum Spanning Trees in linear time, Shortest Paths with Matrix Multiplication.	3
6.	**NP-Hard Problems:** Intro (FPT). Vertex Cover: FPT algorithm, Buss' kernel. Feedback Vertex Set: FPT algorithm, Kernels for Vertex Cover by Matching and for Feedback Vertex Set, Hamiltonian Path Problem, k-Path, Chromatic number, FPT Cut Problems: Important separators, Multiway Cut, Treewidth: Tree decompositions, Algorithmic use (dynamic programming), Introduction to Bidimensionality, Planar Graphs: Linear Kernels, Bidimensionality, Subexponential Time Parameterized Algorithms, Problems on Restricted Graph Classes, Combinatorial Algorithms for Linear Fisher Markets	2
Text Books		
1.	R. Diestel. “Graph Theory”. Springer, 2012.	
2.	Kozen, “Design and Analysis of Algorithms”. Springer	
References		
1.	Douglas B. West, Introduction to Graph Theory, Second Edition, Prentice-Hall	
2.	Bondy, J. A. and Murty, U.S.R., ‘Graph Theory with Applications’, Springer	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Advanced Java	**Semester**		
Department	Computer Science & Engineering	**Course Code**	CST050	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

A student should be able to increase his depth of knowledge in Java programming and explore the uses of the various advanced packages.

Learning Outcomes

At the end of the course the participant will:

- Develop Swing-based GUI
- Develop client/server applications and TCP/IP socket programming
- Update and retrieve the data from the databases using SQL
- Develop distributed applications using RMI
- Develop component-based Java software using JavaBeans
- Develop server side programs in the form of servlets

Course Synopsis

Collection framework, Multithreading, Networking, Enterprise Java Bean, JDBC, Servlets, Java Server Pages, Remote Method Invocation, Common Object Request Broker Architecture.

Course Outline / Content

Unit	Topics	Week
1.	**Collections:** Collection Interfaces, Concrete Collections, The Collections Framework **Multithreading:** Creating thread and running it, Multiple Thread acting on single object, Synchronization, Thread communication, Thread group, Thread priorities, Daemon Thread, Life Cycle of Thread.	2
2.	**Networking:** Internet Addressing, InetAddress, Factory Methods, Instance Methods, TCP/IP Client Sockets, URL, URL Connection, TCP/IP Server Sockets, Datagrams **Enterprise Java Bean:** Preparing a Class to be a JavaBean, Creating a JavaBean, JavaBean Properties, Types of beans, Stateful Session bean, Stateless Session bean, Entity bean.	3
3.	**Java Database Connectivity (JDBC):**Merging Data from Multiple Tables: Joining, Manipulating Databases with JDBC, Prepared Statements, Transaction Processing, Stored Procedures C. **Servlets:** Servlet Overview and Architecture, Interface Servlet and the Servlet Life Cycle, Handling HTTP get Requests, Handling HTTP post Requests, Redirecting Requests to Other Resources, Session Tracking, Cookies, Session Tracking with HttpSession.	3
4.	**JavaServer Pages (JSP):** Introduction, JavaServer Pages Overview, A First JavaServer Page Example, Implicit Objects, Scripting, Standard Actions, Directives, Custom Tag Libraries Remote Method Invocation: Defining the Remote Interface, Implementing the Remote Interface, Compiling and Executing the Server and the Client.	3
5.	**Common Object Request Broker Architecture (CORBA**): Technical/Architectural Overview, CORBA Basics, CORBA services.	3

	Introduction Smart Phone Application Development: Introduction to android platform, Creating application template, adding activity, intent, services to application, receivers and alerts.	
Text Books		
1.	Core and Advanced Java, Black Book, Dreamtech Press	
2.	Java SE8 for Programmers (3rd Edition) (Deitel Developer Series) by Paul Deitel and Harvey Deitel.	
References		
1.	"Advanced Java 2 Platform HOW TO PROGRAM" by H. M.Deitel, P. J. Deitel, S. E. Santry – Prentice Hall	
2.	"Beginning Java™ EE 6 Platform with GlassFish 3 From Novice to Professional" by Antonio Goncalves– Apress publication.	

Department of Computer Science & Engineering National Institute of Technology Srinagar				
Course Title	Numerical Methods	**Semester**		
Department	Computer Science & Engineering	**Course Code**	MTH 707	
Credits	03	**L**	**T**	**P**
Course Type	Theory	3	0	0

Course Objectives

This course is a study of mathematical techniques used to model engineering systems. It involves the development of mathematical models and the application of the computer to solve engineering problems using the following computational techniques: Taylor Series approximation, numerical differentiation, root-finding using bracketing and open methods, linear and polynomial curve fitting, solution methods for matrix equations, numerical integration, and the solution of differential equations

Learning Outcomes

Upon successful completion of this course, the student will:

- Be able to model engineering systems using first and second order differential equations, and solve the equations both analytically and numerically.
- Be able to employ the Taylor Series for approximation and error analysis.
- Be able to formulate and apply numerical techniques for root finding, curve fitting, differentiation, and integration.

Course Synopsis

Numerical analysis is the story of how functions, derivatives, integrals, and differential equations are handled as strings of numbers in the computer. At the heart of numerical analysis is an understanding of the speed of convergence of Taylor, Fourier, and other series expansions. Most scientists and engineers are sooner or later faced with computing tasks that require some knowledge of numerical analysis.

Course Outline / Content

Unit	Topics	Week
1.	Linear Algebra: Matrices, Matrics decomposition: LU decomposition, Cholesky decomposition, spectral decomposition, Matrix Eigen-value problem, Gerchgorin's theorem, Eignen value by iteration, generalized inverse of a matrix, solution of linear system by decomposition method, Jacobi method.	4
2.	Nonlinear system of equations: Newton's method, Powel Hybrid method. Differential equations: Generalised characteristic value problems, phase plane and critical points, stability and phase plane methods in nonlinear equations. Boundary value problems, mixed boundary conditions, boundary conditions at infinity, nonlinear boundary value problems, linear eigen value problems.	4
3.	Differential Equations: Taylor series method, Euler method, Runge-Kutta method. Numerical solutions to Partial Differential Equations: Second order quasi-linear equations, numerical solutions.	3
4.	Approximate Analytic methods: Variational methods, weighted	3

	residual methods – Galerkin's method, collocation method, Functional, quadratic functionals. Numerical Integration – Gauss Legendre, Quadrature, Error Analysis, Convergence of solution. Finite element and Boundary element method.	
Text Books		
1.	S. S. Sastry, 'Numerical Analysis', Prentice-Hall of India Pvt. Ltd.	
2.	M.K. Jain, et.al., 'Numerical Methods for Scientific and Engineering Computation', New Age International Publishers	
References		
1.	Applied Numerical Methods with MATLAB for Engineers and Scientists, 2nd Edition. Stephen C. Chapra, McGraw Hill, 2010	
2.	J. H. Mathews and K. D. Fink, Numerical Methods Using MATLAB®, 3rd ed, Upper Saddle River, NJ: Prentice Hall, 2004, ISBN: 0130652482	
3.	A. Gilat and V. Subramaniam, Numerical Methods for Engineers and Scientists, John Wiley & Sons, Inc., 2008, ISBN: 9780471734406	