SCHEME OF INSTRUCTION AND DETAILED SYLLABI FOR

B. TECH.

IN

DEPARTMENT OF INFORMATION TECHNOLOGY

Effective from 2018 Batch onwards

Dr. B R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY

JALANDHAR – 144011

Phone: 0181-2690301, 02 (Ext. 2502) **Fax: 0181-2690932** **Website: www.nitj.ac.in**

DR B R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY JALANDHAR

INSTITUTE VISION AND MISSION STATEMENTS

VISION

To build a rich intellectual potential embedded with interdisciplinary knowledge, human values and professional ethics among the youth, aspirant of becoming engineers and technologists, so that they contribute to society and create a niche for a successful career.

MISSION

*To become a leading and unique institution of higher learning, offering state-of-the art education, research and training in engineering and technology to students who are able and eager to become change agents for the industrial and economic progress of the nation. To nurture and sustain an academic ambience conducive to the development and growth of committed professionals for sustained development of the nation and to accomplish its integration into the global Econom*y.

DR B R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY JALANDHAR

Program Outcomes (PO)

After completion of the program, students will develop:

PO 1: An ability to apply acquired knowledge of mathematics, science and computer science and engineering to solve the engineering problems.

PO 2: An ability to identify, formulate and analyze engineering problems.

PO 3: An ability to design and implement a system, process, component or program to meet desired needs within realistic constraints such as culture, society, environment health and safety.

PO 4: An ability to conduct investigations of complex problems to reach valid conclusion and to research the contemporary issues.

PO 5: An ability to use appropriate skills, modern tools and techniques necessary for computing and engineering practices.

PO 6: An ability to demonstrate professional responsibilities pertaining to computer science and engineering by the analysis of social health safety legal and cultural issues.

PO 7: An ability to produce engineering solution in global and social context and demonstrate the need for sustainable development.

PO 8: Apply ethical principles, professional ethics and norms of computer engineering practices.

PO 9: An ability to function effectively as an individual and in multi-disciplinary teams.

PO 10: An ability to prepare technical reports and make presentations for the effective delivery of technical information.

PO 11: Recognition of the need for an ability to engage in lifelong learning.

PO 12: An ability to incorporate appropriate economics and business practices for project, risk and change management.

SCHEME OF INSTRUCTION AND DETAILED SYLLABI FOR B. TECH. IN DEPARTMENT OF INFORMATION TECHNOLOGY

THIRD SEMESTER

S.No.	Sub. Code	Subjects	L	T	P	Credits	
1	ITPC-201	Object Oriented Programming Concepts	3	0	0	3	PC
2	ITPC-203	Data Structures	3	0	0	3	PC
3	ITPC-205	Data Communication and Networking	3	0	0	3	PC
4	ITPC-207	Fundamentals of Database Management Systems	3	0	0	3	PC
5	ITPC-209	Computer System Architecture	3	1	0	4	PC
6	MACI-203	Numerical Methods	3	1	0	4	CIC
Practical /Training /Projects							
7	ITPC-221	Object Oriented Programming Concepts Lab	0	0	2	1	PC
8	ITPC-223	Data Structures Lab	0	0	2	1	PC
9	ITPC-225	Data Communication and Networking Lab	0	0	2	1	PC
10	ITPC-227	Fundamentals of Database Management Systems Lab	0	0	2	1	PC
TOTAL			**18**	**2**	**8**	**24**	

FOURTH SEMESTER

S.No.	Sub. Code	Subjects	L	T	P	Credits	
1	ITPC-202	Introduction to Design and Analysis of Algorithms	3	0	0	3	PC
2	ITPC-204	Operating System Concepts	3	0	0	3	PC
3	ITPC-206	Java Programming	3	0	0	3	PC
4	ITPC-208	Data Mining and Data Warehousing Concepts	3	0	0	3	PC
5	ITPC-210	Formal Language and Automata Theory	3	1	0	4	PC
6	HMCI-201	Economics for Engineers	3	0	0	3	CIC
Practical /Training /Projects							
7	ITPC-222	Introduction to Design and Analysis of Algorithms Lab	0	0	2	1	PC
8	ITPC-224	Operating System Concepts Lab	0	0	2	1	PC
9	ITPC-226	Java Programming Lab	0	0	2	1	PC
10	ITPC-228	Data Mining and Data Warehousing Concepts Lab	0	0	2	1	PC
TOTAL			**18**	**1**	**8**	**23**	

Fifth Semester

S.No.	Sub. Code	Subjects	L	T	P	Credits	
1	ITPC-301	Cryptography and Network Security	3	0	0	3	PC
2	ITPC-303	Software Engineering Concepts	3	0	0	3	PC
3	ITPC-305	Web Design Technologies	3	0	0	3	PC
4	ITPC-307	Computer Graphics and Applications	3	0	0	3	PC
5	ITPC-309	Discrete Mathematics	3	1	0	4	PC
6	ITPE-XXX	DE –I	3	0	0	3	PE
Practical /Training /Projects							
7	ITPC-321	Cryptography and Network Security Lab	0	0	2	1	PC
8	ITPC-323	Software Engineering Concepts Lab	0	0	2	1	PC
9	ITPC-325	Web Design Technologies Lab	0	0	2	1	PC
10	ITPC-327	Computer Graphics and Applications Lab	0	0	2	1	PC
11	ITCI-300	Minor Project Phase-I	0	0	2	0*	CIC
TOTAL			**18**	**1**	**10**	**23**	

SIXTH SEMESTER

S.No.	Sub. Code	Subjects	L	T	P	Credits	
1	ITPC-302	Soft Computing Concepts	3	0	0	3	PC
2	ITPC-304	Object-Oriented Modeling and Design with UML	3	0	0	3	PC
3	ITPC-306	Mobile Application Development	3	0	0	3	PC
4	ITPC-308	Machine Learning Concepts	3	0	0	3	PC
5	ITPE-XXX	DE-II	3	0	0	3	PE
6	ITOE-XXX	OE -I	3	0	0	3	OE
Practical /Training /Projects							
7	ITPC-322	Soft Computing Concepts Lab	0	0	2	1	PC
8	ITPC-324	Object-Oriented Modeling and Design with UML Lab	0	0	2	1	PC
9	ITPC-326	Mobile Application Development Lab	0	0	2	1	PC
10	ITCI-301	Minor Project Phase-II	0	0	2	2*	CIC
TOTAL			**18**	**0**	**8**	**23**	

SEVENTH SEMESTER

S.No.	Sub. Code	Subjects	L	T	P	Credits	
1	ITPC-401	Software Testing	3	0	0	3	PC
2	ITPC-403	Cloud Computing	3	0	0	3	PC
3	ITPE-XXX	DE – III	3	0	0	3	PE
4	ITPE-XXX	DE – IV	3	0	0	3	PE
5	ITOE-XXX	OE – II	3	0	0	3	OE
Practical /Training /Projects							
6	ITPC-421	Software Testing Lab	0	0	2	1	PC
7	ITPC-423	Cloud Computing Lab	0	0	2	1	PC
8	ITCI-401	Industrial Practical Training	0	0	8	2	CIC
9	ITCI-400	Major Project (Phase I)	0	0	4	0*	CIC
TOTAL			**15**	**0**	**16**	**19**	

EIGHTH SEMESTER

S.No.	Sub. Code	Subjects	L	T	P	Credits	
1	ITPC- 402	Introduction to System Programming	3	0	0	3	PC
2	ITPC- 404	E- Commerce	3	0	0	3	PC
3	ITPC- 406	Decision Support Systems Methodology	3	0	0	3	PC
4	ITPE-XXX	DE – V	3	0	0	3	PE
5	ITOE-XXX	OE – III	3	0	0	3	OE
Practical /Training /Projects							
6	ITPC-422	Introduction to System Programming Lab	0	0	2	1	PC
7	ITCI-402	Industrial Lecture	0	0	2	1	CIC
8	ITCI-400	Major Project (Phase II)	0	0	8	4*	CIC
TOTAL			**15**	**0**	**12**	**21**	

LIST OF DEPARTMENTAL ELECTIVES

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITPE-051	Advanced Concepts in Operating System	3	0	0	3
2	ITPE-052	Wireless Data Networks	3	0	0	3
3	ITPE-053	Information Security System	3	0	0	3
4	ITPE-054	Mobile Computing	3	0	0	3
5	ITPE-055	Software Project Management Concepts	3	0	0	3
6	ITPE-056	Agile Software Development	3	0	0	3
7	ITPE-057	Principles of Compiler Design	3	0	0	3
8	ITPE-058	Principles of Programming Languages	3	0	0	3
9	ITPE-059	Data Analytics	3	0	0	3
10	ITPE-060	Digital Image Processing	3	0	0	3
11	ITPE-061	Multicore Programming	3	0	0	3
12	ITPE-062	Cyber Forensic	3	0	0	3
13	ITPE-063	Artificial Intelligence Concepts	3	0	0	3
14	ITPE-064	Internet of Things (IoT) Concepts	3	0	0	3
15	ITPE-065	Mobile Database Management System	3	0	0	3
16	ITPE-066	Advanced Computer Networks	3	0	0	3
17	ITPE-067	Fundamental of Blockchain Technology	3	0	0	3
18	ITPE-068	High Performance Computing	3	0	0	3
19	ITPE-069	Advanced Computer Architecture	3	0	0	3
20	ITPE-070	Natural Language Processing	3	0	0	3

LIST OF OPEN ELECTIVES

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITOE-001	Fundamentals of Software Engineering	3	0	0	3
2	ITOE-002	Web Design Concepts	3	0	0	3
3	ITOE-003	Fundamentals of Data Analytics	3	0	0	3
4	ITOE-004	Agile Software Engineering	3	0	0	3
5	ITOE-005	Mobile Application Development Concepts	3	0	0	3
6	ITOE-006	Fundamentals of Cloud Computing	3	0	0	3
7	ITOE-007	Machine Learning	3	0	0	3
8	ITOE-008	Fundamentals of Artificial Intelligence	3	0	0	3

Summary Sheet of Credits

Semester	Course Category	Number	Credits	Total Credits
III	CIC	1	4	24
	PC	5+4*=09	20	
	PE	-	-	
	OE	-	-	
IV	CIC	01	03	23
	PC	5+4*=09	20	
	PE	-	-	
	OE	-	-	
V	CIC	1	0	23
	PC	5+4*=09	20	
	PE	1	3	
	OE	-	-	
VI	CIC	1	2	23
	PC	4+3*=7	15	
	PE	1	3	
	OE	1	3	
VII	CIC	2	2	19
	PC	2+2*=4	8	
	PE	2	6	
	OE	1	3	
VIII	CIC	2	5	21
	PC	3+1*=4	10	
	PE	1	3	
	OE	1	3	
Total Credits (III to VIII Sem)				133
Total No of PC		42	93	
Total No of PE		05	15	
Total No of OE		03	09	
Total Credits of CIC		07	16	
Credits from 1st year			47	47
Total Credits			180	180

THIRD SEMESTER

S.No.	Sub. Code	Subjects	L	T	P	Credits	
1	ITPC-201	Object Oriented Programming Concepts	3	0	0	3	PC
2	ITPC-203	Data Structures	3	0	0	3	PC
3	ITPC-205	Data Communication and Networking	3	0	0	3	PC
4	ITPC-207	Fundamentals of Database Management Systems	3	0	0	3	PC
6	ITPC-209	Computer System Architecture	3	1	0	4	PC
5	MACI-203	Numerical Methods	3	1	0	4	CIC
Practical /Training /Projects							
7	ITPC-221	Object Oriented Programming Concepts Lab	0	0	2	1	PC
8	ITPC-223	Data Structures Lab	0	0	2	1	PC
9	ITPC-225	Data Communication and Networking Lab	0	0	2	1	PC
10	ITPC-227	Fundamentals of Database Management Systems Lab	0	0	2	1	PC
TOTAL			**18**	**2**	**8**	**24**	

Course Code:	**ITPC-201**
Subject Name	**OBJECT ORIENTED PROGRAMMING CONCEPTS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objectives

- Able to differentiate between structure oriented programming and object oriented programming and to specify simple abstract data types and design implementations
- Recognize features of object oriented design such as encapsulation, polymorphism, inheritance and composition of systems based on object identity.
- Able to use object oriented programming language like C++ and associated library to develop object oriented programs.

Course Outcomes

1. Develop skill to identify and determine the usage of various data structures, operations, associated algorithms and implement their applications.
2. Apply knowledge of pointers, memory allocation and string handling for solving programming problems.
3. Understand the concept of trees and graphs, their implementation and applications.
4. Able to implement standard algorithms for searching and sorting.
5. Analyze efficiency of different algorithms using time and space complexity.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-201												
CO 1	M	M	H	L	L	L				M		
CO 2	M	M	H	L	L					M		
CO 3	M	L	H	M	L		M			M		
CO 4	L	M	H	M					M	L		
CO 5		L	H	L				M	M			

Course Content

Object oriented thinking: Need for OOP Paradigm, Procedural programming Vs object oriented programming, object oriented concepts.

Functions: Main function, function prototyping, inline functions, reference variables, call by reference, Defaults arguments, function overloading, Math library functions.

Class: Difference between C structure and class, specifying a class, Defining member functions: inside and outside class, scope resolution operator, Array within a class, array of objects, Static data members and member functions, Object as function arguments, returning objects, Friend function, memory allocation for objects, pointer to members, pointer to object, this pointer local classes.

Constructor and destructor: Constructor, types of constructors: default, parameterized and copy constructor, constructor overloading, constructor with default parameter, dynamic initialization of objects, destructor

Operator overloading and Type Conversion: Defining operator overloading, overloading unary and binary operator, Data Conversion: Basic to User Defined, User defined to basic, Conversion from one user-defined type to another.

Inheritance and polymorphism: Base class, derived class, visibility modes, derivation and friendship, Types of inheritance, Containership, virtual function binding, pure virtual functions, Abstract class, pointer to derived class.

Console IO operations: C++ stream classes, Unformatted IO operations, formatted IO operations, managing output with manipulators.

Working with files: Classes for file stream operations, opening and closing files, detectinf cof, File opening modes, file Pointers, Error handling during file operations, command line arguments. Templates: Class template, class template with parameter, function template, function template with parameter.

Recommended Books

1. Bjrane Stroustrup, "C++ Programming Language", 3rd Ed., Pearson education Asia, 2002.
2. Lafore R."Object Oriented Programming in C++",4th Ed. Tech media, New Delhi, 2002.
3. Yashwant Kenetkar,"Let us C++",1st Ed., Oxford University Press, 2006.
4. B.A. Forouzan and R.F. Gilberg, "Computer Science: A structured approach using C++" 2nd Ed. Cengage Learning, New Delhi, 2004.
5. E. Balagurusamy, "Object Oriented Programming with C++" 5th Ed., Tata McGraw Hill, 2011.

Course Code:	**ITPC-203**
Subject Name	**DATA STRUCTURES**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objectives

- This course introduces the concept of Data Structures used in various computer science applications.
- The students are introduced to understand and efficiently apply various data structures such as stacks, queues, linked lists, trees and graphs for solving various computing problems using C programming language.

Course Outcomes

1. Develop skill to identify and determine the usage of various data structures, operations, associated algorithms and implement their applications.
2. Apply knowledge of pointers, memory allocation and string handling for solving programming problems.
3. Understand the concept of trees and graphs, their implementation and applications.
4. Able to implement standard algorithms for searching and sorting.
5. Analyze efficiency of different algorithms using time and space complexity.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-203												
CO 1	H	M	M	M	M							
CO 2		H	M	M	M	M	M		L			
CO 3	L	M	M	H	M							
CO 4		H	H	L								
CO 5	M	M	H	H	M	H	M		H		H	

Course Content

Introduction: Basic Terminology, Elementary Data Organization, Algorithm, Efficiency of an Algorithm, Time and Space Complexity, Asymptotic notations: Big-Oh, Time-Space trade-off. Abstract Data Types (ADT)

Arrays: Definition, Single and Multidimensional Arrays, Representation of Arrays: Row Major Order, and Column Major Order, Application of arrays, Sparse Matrices and their representations.

Linked lists: Array Implementation and Dynamic Implementation of Singly Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List. Insertion, Deletion, Traversal, Polynomial Representation and Addition, Generalized Linked List

Stacks: Abstract Data Type, Primitive Stack operations: Push & Pop, Array and Linked Implementation of Stack in C, Application of stack: Prefix and Postfix Expressions, Evaluation of postfix expression, Recursion, Tower of Hanoi Problem, Simulating Recursion, Principles of recursion, Tail recursion, Removal of recursion Queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, Array and linked implementation of queues in C, Dequeue and Priority Queue.

Trees: Basic terminology, Binary Trees, Binary Tree Representation: Array Representation and Dynamic Representation, Complete Binary Tree, Algebraic Expressions, Extended Binary Trees, Array and Linked Representation of Binary trees, Tree Traversal algorithms: Inorder, Preorder and Postorder, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm.

Graphs: Terminology, Sequential and linked Representations of Graphs: Adjacency Matrices, Adjacency List, Adjacency Multi list, Graph Traversal : Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transistive Closure and Shortest Path algorithm: Warshal Algorithm and Dijikstra Algorithm, Introduction to Activity Networks.

Searching : Sequential Search, Binary Search, Comparison and Analysis Internal Sorting: Insertion Sort, Selection, Bubble Sort, Quick Sort, Two Way Merge Sort, Heap Sort, Radix Sort, Practical consideration for Internal Sorting.

Search Trees: Binary Search Trees (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, AVL trees, Introduction to m-way Search Trees, B Trees & B+ Trees

Hashing: Hash Function, Collision Resolution Strategies Storage Management: Garbage Collection and Compaction.

Recommended Books

1. Schaum Outline series by: Seymour Lipschutz, “Data Structures”, Publishers: Tata McGraw Hill, New Delhi, Year of Publication: 2015.
2. M. Tenenbaum, “Data Structures using C & C++”, Prentice-Hall of India Pvt. Ltd., New Delhi, Year of Publication:2012.
3. E. Horowitz, S. Sahni, S.Anderson-freed, “Fundamentals of Data Structures in C”, Second Edition, University Press, 2008.
4. R. Kruse and C.L. Tondo, “Data Structures and Program Design in C”, 2nd Edition,1996, Pearson, Year of Publication: 2006
5. Gilberg Forozan, “Data Structure – A pseudo code approach with C++”, Cengage Learning, New Delhi. Year of Publication: 2001

Course Code:	**ITPC-205**
Subject Name	**DATA COMMUNICATION AND NETWORKING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Understand computer network basic, different models used for study of computer networks, ability to identify different designs, understanding of the issues surrounding wired and wireless Networks.
- Design, calculate, and apply subnet masks to fulfill networking requirements and building the skills of routing mechanisms.
- Analyze the features and operations of various application layer protocols such as Http, DNS, SMTP and FTP.
- Analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies
- Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Course Outcomes

1. Build an understanding of the fundamental concepts of computer networking.
2. Familiarize the student with the basic taxonomy and terminology of the computer networking area. Introduce the student to advanced networking concepts, preparing the student for entry into advanced courses in computer networking.
3. Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-205												
CO 1	H		M									
CO 2	M	H										
CO 3		M	M									

Course Content

Data Transmission/The Physical Layer: Concepts and Terminology, Analog and Digital Data Transmission, Transmission Impairments, Guided Transmission Media, Wireless Transmission, Communication Satellites, The Public Switched Telephone Network, The Mobile Telephone System, Cable Television

Data Encoding: Digital Data: Digital and Analog Signals, Analog Data: Digital and Analog Signals, Spread Spectrum

Data Communication Interface: Asynchronous and Synchronous Transmission, Line Configurations, Interfacing

Multiplexing: Frequency-Division Multiplexing, Synchronous Time-Division Multiplexing, Statistical Time-Division Multiplexing

Circuit Packet and Switching: Switched Networks, Circuit-Switching Networks, Switching Concepts, Routing in Circuit-Switched Networks, Control Signaling, Packet-Switching Principles, Routing, Congestion Control, X.25 282

Frame Relay: Frame Relay Protocol Architecture, Frame Relay Call Control, User Data Transfer, Network Function, Congestion Control

LAN Technology and Systems: LAN Architecture, BusITree LANs, Ring LANs, Star LANs, Wireless LANs, Ethernet and Fast Ethernet (CSMAICD), Token Ring and FDDI, 100VG-AnyLAN, ATM LANs, Fibre Channel, Wireless LANs, Bridge Operation, Routing with Bridges

Protocols and Architecture: Protocols, OSI, TCP/IP Protocol Suite

Examples of networks: Novell Netware, Arpanet, and Internet. Examples of Data Communication Services: X.25 Networks, Frame relay, Broad band ISDN and ATM. Physical Layer: Transmission media- Narrow band ISDN: Services-Architecture- Interface, Broad band ISDN and ATM- Virtual Circuits versus Circuit Switching –Transmission in ATM networks. FDDI

Link Layer and Local Area Networks Data link layer: Service provided by data link layer-Error detection and correction Techniques-Elementary data link layer protocols -Sliding Window protocols - Data link layer in HDLC, Internet and ATM . Multiple Access protocols: Channel partitioning protocols: TDM-FDM-Code Division Multiple Access(CDMA) .Random Access protocols : ALOHA CSMA and CSMA/CD . Local area Network: LAN addresses- Address Resolution Protocol-Reverse Address Resolution Protocol. Ethernet: Ethernet Technologies-IEEE standards- Hubs-Bridges and Switches

Recommended Books

1. Andrew S. Tanenbaum, “Computer Networks” Ed Pearson Education 4th Edition, 2010.
2. James F. Kurose and Keith W. Ross, “Computer Networking” Pearson Education, 2012.
3. William Stalling, “Data and Computer Communication”, Pearson Education, 7th Edition, 2nd Indian Reprint 2010.
4. Behrouz A. Fourouzan, Data Communications and Networking, 2/e Tat McGraw-Hill, 2006.

Course Code:	**ITPC-207**
Subject Name	**FUNDAMENTALS OF DATABASE MANAGEMENT SYSTEMS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- To understand fundamentals of data models and to conceptualize a database system for user requirement.
- To study the fundamental of Database query language, like SQL and relational algebra.
- To learn the concept of normalization in database design.
- To learn fundamental concepts of transaction processing, concurrency control techniques and database recovery procedure.
- Understand the professional, ethical and security issues and responsibilities in database design.

Course Outcomes

1. To understand fundamental concepts, principles and applications of database system.
2. To demonstrate database related programming languages and perform the basics of commercial relational systems.
3. To apply the concepts of normalization on database design.
4. To Design and Implement a small database project, considering the issues like concurrency control, recovery and security.
5. Comprehend contemporary issues related to industry and government related to database domain.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-207												
CO 1	M		H		M	L						
CO 2			H	L	M							
CO 3			M	M	M	H	M	M	M			
CO 4		M	H	L	H	L		H	M			M
CO 5						M						

Course Content

Introduction: An overview of database management system, database system Vs file system, Database system concepts and architecture, data models schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure.

Data Modeling using the Entity Relationship Model: ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationships of higher degree.

Relational data Model and Language: Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, tuple and domain calculus.

Introduction to SQL: Characteristics of SQL. Advantage of SQL. SQL data types and literals. Types of SQL commands. SQL operators and their procedure. Tables, views and indexes. Queries and sub queries. Aggregate functions. Insert, update and delete operations. Joins, Unions, Intersection, Minus, Cursors in SQL.

Data Base Design & Normalization: Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependences, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

Transaction Processing Concepts: Transaction system, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling.

Crash Recovery: Failure classification, recovery concepts based on deferred update, recovery concepts based on intermediate update, shadow paging, check points, on-line backup during database updates

Concurrency Control Techniques: Concurrency control, locking Techniques for concurrency control, Time stamping protocols for concurrency control, validation based protocol, multiple granularity, Multi version schemes, Recovery with concurrent transaction.

Client/Server Databases: Client/Server concepts, approach, Client/Server environments, characterization of Client/Server computing, application partitioning, the two-layer, and the Three layer architecture, Client/Server communication, APIs in Client/Server computing, middleware technology, application developments, design concepts, Client application development tools, and database servers.

Integrity, Security and Repositories: Needs for database integrity, integrity constraints, non-procedural integrity constraints, integrity constraints specifications in SQL, introduction to database security mechanism, security specification in SQL, system catalogues

Case Studies: Case studies on databases such as Oracle, MySQL etc.

Recommended books:

1. Date C J, “An Introduction To Database System”, 8th Ed., Addision Wesley, 2003.
2. Korth, Silbertz, Sudarshan, “Database Concepts”, 6th Ed., McGraw Hill, 2010.

3. Elmasri, Navathe, “Fundamentals Of Database Systems”, 5th Ed., Addision Wesley, 2007.
4. Bipin C. Desai, “An introduction to Database Systems”, 8th Ed., Galgotia Publication, 2006.
5. Rob and Coronel, “Database Systems” 5th Ed.,Cengage Learning, 2008.

Course Code:	**ITPC-209**
Subject Name	**COMPUTER SYSTEM ARCHITECTURE**
Contact hours/Credit Scheme (L-T-P-C)	**3-1-0-4**
Pre-requisites	**NONE**

Course Objectives

- Working of Computer Systems & its basic principles
- Concepts of processor and control design.
- Concepts of pipelining techniques.
- Memory hierarchy and its organization.
- Concept of I/O devices and working.

Course Outcomes

1. Understand the Computer System concepts.
2. Understand the organization of a computer system in terms of its main components
3. Understand the processor and control design of a computer system.
4. Understand the various types of memory.
5. Understand input/output mechanisms.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-209												
CO 1	M	L	M	H			M					
CO 2	M	L	L	M	M	M	H					
CO 3	H			M		L					H	
CO 4	L	M				H					M	
CO 5			L	H	M	M				M	M	

Course Content

Introduction: Number representation; fixed and floating point number representation, IEEE standard for floating point representation. Error detection and correction codes: Hamming code.

Digital computer generation, computer types and classifications, functional units and their interconnections, buses, bus architecture, types of buses and bus arbitration. Register, bus and memory transfer.

Central Processing Unit: Addition and subtraction of signed numbers, look ahead carry adders. Multiplication: Signed operand multiplication, Booths algorithm and array multiplier. Division and logic operations. Floating point arithmetic operation, Processor organization, general register organization, stack organization and addressing modes.

Control Unit: Instruction types, formats, instruction cycles and subcycles (fetch and execute etc) , micro-operations, execution of a complete instruction.

Hardwire and micro programmed control: microprogramme sequencing, wide branch addressing, microinstruction with next address field, pre-fetching microinstructions, concept of horizontal and vertical microprogramming.

Memory: Basic concept and hierarchy, semiconductor RAM memories, 2D & 2 1/2D memory organization. ROM memories. Cache memories: concept and design issues 9 performance, address mapping and replacement) Auxiliary memories: magnetic disk, magnetic tape and optical disks Virtual memory: concept implementation.

Input / Output: Peripheral devices, I/O interface, I/O ports, Interrupts: interrupt hardware, types of interrupts and exceptions. Modes of Data Transfer: Programmed I/O, interrupt initiated I/O and Direct Memory Access., I/O channels and processors. Serial Communication: Synchronous & asynchronous communication, standard communication interfaces.

Recommended books

1. M Moris Mano, "Computer System Architecture", Pearson Education, 3rd Ed.
2. David A. Patterson and John L. Hennessy, "Computer Organization & Design-The Hardware/Software Interface", Morgan Kaufmann, 2nd Ed.
3. William Stallings, "Computer Organisation and Architecture, Designing for Performance", Pearson Education Asia, 6th Ed. 2003.
4. Harry F. Jordan and Gita Alaghband, "Fundamentals of Parallel Processing", Pearson Education, 1st Ed. 2003.
5. Barry Wilkinson Michael Allen, "Parallel Programming", Prentice Hall.
6.

Course Code:	**MACI-203**
Subject Name	**NUMERICAL METHODS**
Contact hours/Credit Scheme (L-T-P-C)	**3-1-0-4**
Pre-requisites	**NONE**

COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

COURSE OUTCOMES

After the course completion, the student will be able to

1. Understand the different numerical methods to solve the algebraic equations and to solve system of linear and non linear equations.
2. Understand the different numerical methods for interpolation, differentiation, integration and solving set of ordinary differential equations.
3. Understand how numerical methods afford a mean to generate solutions in a manner that can be implemented on digital computers.

Course Outcomes	Program outcomes										
									P	P	P

MACI-203												
CO 1.												
CO 2.			L									
CO 3.												

TOPICS COVERED

Roots of algebraic and transcendental equations, Bisection method, Regula – Falsi method, Newton – Raphson method, Bairstow's method and Graeffe's root squaring method.

Solution of simultaneous algebraic equations, matrix inversion and eigen-value problems, triangularisation method, Jacobi's and Gauss-Siedel iteration method, partition method for matrix inversion, power method for largest eigen-value and Jacobi's method for finding all eigen-values.

Finite differences, interpolation and numerical differentiation, forward, backward and central differences, Newton's forward, backward and divided difference interpolation formulas, Lagrange's interpolation formula, Stirling's and Bessel's central difference interpolation formulas, numerical differentiation using Newton's forward and backward difference formulas and numerical differentiation using Stirling's and Bessel's central difference interpolation formulas.

Numerical integration, Trapezoidal rule, Simpson's one-third rule and numerical double integration using Trapezoidal rule and Simpson's one-third rule.

Taylor's series method, Euler's and modified Euler's methods, Runge-Kutta fourth order methods for ordinary differential equations, simultaneous first order differential equations and second order differential equations.

Boundary value problems, finite difference methods for boundary value problems.

Partial differential equations, finite difference methods for elliptic, parabolic and hyperbolic equations.

TEXT BOOKS, AND/OR REFERENCE MATERIAL

1. S SSastry, Introductionary Methods of Numerical Analysis, 3rd Edition, Prentice Hall of India Pvt.Ltd., New India -1999
2. S C Chapra and R P Canale, Numerical Methods for Engineers, 2nd Edition, McGraw Hill Book Company, Singapore 1990.
3. Kalavathy S., "Numerical Methods", Cengage Publishers, New Delhi.
4. Burden Richard L. ,Faires J. Douglas, "Numerical Anlaysis" , Cengage Learning , New Delhi.

Course Code:	**ITPC-221**
Subject Name	**OBJECT ORIENTED PROGRAMMING CONCEPTS LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

- To make the student learn an object oriented way of solving problems.
- To teach the student to write programs in C++ to solve the problems.

Course Outcomes

1. Understand the features of C++ supporting object oriented programming
2. Understand the relative merits of C++ as an object oriented programming language
3. Understand how to produce object-oriented software using C++
4. Understand how to apply the major object-oriented concepts to implement object oriented programs in C++, encapsulation, inheritance and polymorphism
5. Understand advanced features of C++ specifically stream I/O, templates and operator overloading

List of practicals

1. Program to Implement Various Control Structures: If statement, Switch case statement, do while loop, For loop, While loop
2. Program to make the use of inline function.
3. Programs to demonstrate the concept of Structure & Unions
4. Programs to implement the concept of Pointer Arithmetic.
5. Program to implement the concept of Functions & Recursion
6. Programs to Understand Different Function Call Mechanism
7. Programs to Understand Storage Specifiers,
8. Program to understand the Use of “this” Pointer.
9. Program to demonstrate the concept of:
   a) Default constructor
   b) Parameterized constructor
   c) Copy constructor
   d) Constructor overloading
10. Program to demonstrate the concept of destructor.
11. Program to show multiple inheritance
12. Program to show multilevel inheritance
13. Program to show hybrid inheritance
14. Program to show the concept of containership.
15. Program to overload unary operator.
16. Program to overload binary operator
17. Program to show the concept of run time polymorphism using virtual function.
18. Program to work with formatted and unformatted IO operations.
19. Program to copy one file onto the end of another, adding line numbers

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code:	**ITPC-223**
Subject Name	**DATA STRUCTURE LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

- To understand how the choice of data structures can lead to efficient implementations of algorithms.
- To familiarize how certain applications can benefit from the choice of data structures.

Course outcomes

1. Understand the use of arrays, records, linked structures, stacks, queues, trees, and graphs.
2. Understand how the choice of data structures can lead to efficient implementations of algorithms.
3. Familiarize how certain applications can benefit from the choice of data structures.
4. Apply and implement the learned algorithm for problem solving

LIST OF PRACTICALS

1. Implementation of various operations on array like Print the elements of an array in reverse order, delete an element from an array at a given index, find the smallest element in an array, calculate the sum of elements in an array etc.
2. Implementation of linear and binary search technique in an array of n values.
3. Implementation of sparse matrix and perform various operations like search the sparse matrix for an element specified by the user. Print the result of the search appropriately. Use the triple <row, column, value> to represent an element in the sparse matrix.
4. Implementation of various operations on linked list: Creation, Insertion, Deletion, Reverse etc.
5. Implementation of various operations on doubly linked list: Creation, Insertion, Deletion etc.
6. Implementation of stack data structure using array of n values and perform push () and pop () operation.
7. Implementation of stack operations to convert a given infix expression into its equivalent postfix expression. Implement the stack using an array.
8. Implementation of queue data structure using array and linked list with the basic functions of Create(), IsEmpty(), Insert(), Delete() and IsFull().
9. Implementation of circular queue using an array with the following operations: a) Insert a new element b) Delete a given element c) Display the content of queue
10. Implementation of a double ended queue using i) array and ii) doubly linked list respectively.
11. Implementations of Binary Tree menu driven program :
    a. Traverse the tree using all the methods i.e. inorder, preorder & postorder to display the elements in the tree
    b. Copying tree
    c. Counting the number of nodes in the tree
    d. Counting only leaf nodes in the tree.
12. Implementations of BST program with at least following operations.
    a) To construct a binary search tree of n values.
    b) To insert new elements in BST
    c) To delete elements in BST at various positions
13. Implementation of the following sorting methods to arrange a list of n values in ascending order:
    a) Insertion sort b) Merge sort c) Selection sort d) Bubble sort e) Quick sort f) Heap Sort
14. Implementation of the graph traversal algorithms: a) Depth first traversal b) Breadth first traversal
15. Implementation of Minimal Spanning Trees.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code:	**ITPC-225**
Subject Name	**DATA COMMUNICATION AND NETWORKING LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives

- To understand the working of the fundamental elements of communications systems.
- Study of different modulation techniques.

Course Outcomes

1. Understand basic computer network technology, data communications System and its components.
2. Identify the different types of network topologies and protocols, to enumerate the layers of the OSI model and TCP/IP
3. Identify the different types of network devices and their functions within a network.
4. Understand and build the skills of subnetting and be familiar with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

List of Practicals

1. Network Physical Components Hands-on (Networks Cabling)
2. Brief introduction to Cisco Packet Tracer covering network devices, cables and end-devices. Connect one PC to another PC using the cable and also design a network of few computers using hub/switch and assign IP address and subnet mask to them. Implement Star topologies in Packet Tracer.
3. Implement Bus, Ring, Mesh, and Hybrid topology. Assign IP address and subnet mask to each computer and run the ping command to check the reachability of the systems. Send message between source and destination and observe the flow of the messages.
4. Design two separate network and connect them using Router
5. Implementation and performance of STOP and Wait protocol
6. Implementation and performance of Sliding Window protocol
7. Design Static route configuration, Dynamic route configuration, and Default route configuration
8. Design NAT and PAT
9. Design VLAN
10. Experiencing Real-world Network Infrastructure
11. Experiencing Real-world network Devices using Switches and Routers
12. Implementing Client/Server Applications
13. Packet Capturing and Analysis using the monitoring tool
14. Application test using the network monitoring tool

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents

Course Code:	**ITPC-227**
Subject Name	**FUNDAMENTALS OF DATABASE MANAGEMENT SYSTEMS LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

- To provide a strong formal foundation in database concepts, technology and practice to the participants to groom them into well-informed database application developers.

Course outcomes

1. To describe data models and schemas in DBMS
2. To understand the features of database management systems and Relational database.
3. To use SQL- the standard language of relational databases.
4. To understand the functional dependencies and design of the database.
5. To understand the concept of Transaction and Query processing.

List of Practicals

1. Perform Data Definition Language (DDL) commands in RDBMS
2. Perform Data Manipulation Language (DML) and Data Control Language (DCL)
3. High level language extensions with cursors such as student Grade Calculation and Bill Calculation.
4. Triggers using High level language extension such as Display student result and invalid condition.
5. Procedures and Functions such as insert number and find factorial.
6. Embedded SQL like bonus calculation.
7. Database design using E-R model and Normalization
8. Design and implementation of payroll processing system (Implement and perform)
9. Design and implementation of Banking system
10. Design and implementation of Library Information System
11. Design and implementation of Student Information System
12. Automatic Backup of Files and Recovery of Files

* Students are advised to use **Developer 2000/Oracle-10i** or higher version or other latest version for above listed experiments. Mini Project may also be planned & carried out throughout the semester to understand the important various concepts of Database.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

FOURTH SEMESTER

S.No.	Sub. Code	Subjects	L	T	P	Credits	
1	ITPC-202	Introduction to Design and Analysis of Algorithms	3	0	0	3	PC
2	ITPC-204	Operating System Concepts	3	0	0	3	PC
3	ITPC-206	Java Programming	3	0	0	3	PC
4	ITPC-208	Data Mining and Data Warehousing Concepts	3	0	0	3	PC
5	ITPC-210	Formal Language and Automata Theory	3	1	0	4	PC
6	HMCI-201	Economics for Engineering	3	0	0	3	CIC
Practical /Training /Projects							
7	ITPC-222	Introduction to Design and Analysis of Algorithms Lab	0	0	2	1	PC
8	ITPC-224	Operating System Concepts Lab	0	0	2	1	PC
9	ITPC-226	Java Programming Lab	0	0	2	1	PC
10	ITPC-228	Data Mining and Data Warehousing Concepts Lab	0	0	2	1	PC
TOTAL			**18**	**1**	**8**	**23**	

Course Code:	**ITPC-202**
Subject Name	**INTRODUCTION TO DESIGN AND ANALYSIS OF ALGORITHMS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- To develop the understanding of advanced data structures.
- To develop the understanding of algorithmic design paradigms.

Course Outcomes

1. Compare, contrast, and apply the key algorithmic design paradigms: brute force, divide and conquer, greedy, dynamic.
2. Compare, contrast, and apply key data structures: trees, lists, stacks, queues, hash tables, and graph representations.
3. Compare, contrast, and apply algorithmic tradeoffs: time vs. space, deterministic vs. randomized, and exact vs. approximate.
4. Implement, empirically compare, and apply fundamental algorithms and data structures to real-world problems.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-202												
CO 1	M	M	H	M	L							
CO 2					M	H						
CO 3	L	M		M			L		M			
CO 4			M		L						M	

Course Content

Introduction: Growth of Functions, analyzing algorithms, complexity of algorithms, Growth of functions, asymptotic notation, standard notations and common functions, The substitution method, The recursion-tree method.

Sorting and order Statistics: Insertion sort, Quick Sort, Merge Sort, Heap Sort, Comparison of sorting algorithms, Sorting in linear time - Counting sort, Radix sort, Bucket sort, medians and order statistics.

Advanced Data Structures*:* Red-Black trees, Heaps, Hash tables, Binomial Heaps, Fibonacci Heaps.

Divide and Conquer: Divide and Conquer with examples such as Sorting, Matrix Multiplication.

Dynamic Programming: Elements of dynamic programming, Assembly-line scheduling, Matrix-chain multiplication, Longest common subsequence, Optimal binary search trees

Greedy Algorithms: Elements of the greedy strategy, Greedy methods with examples such as activity-selection problem, Huffman codes, task-scheduling problem

Minimum Spanning trees: Graph Theory, Growing a minimum spanning tree, Prim’s and Kruskal’s algorithms.

Single source and All-Pairs shortest paths: Dijkstra’s and Bellman Ford algorithms, Shortest paths and matrix multiplication, The Floyd-Warshall algorithm, Johnson's algorithm for sparse graphs

Linear Programming: Standard and slack forms, Formulating problems as linear programs, The simplex algorithm, Duality, The initial basic feasible solution

Branch and Bound with examples such as Travelling Salesman Problem, Graph Coloring

Overview of NP-Completeness: Polynomial time, NP-completeness, NP-complete problems

Recommended books:

1. Cormen, Leiserson and Rivest , "Introduction to Algorithms", McGraw-Hill, 3^{rd} Ed.
2. Ullman and Hopcroft, "Design and Analysis of Algorithms", Pearson Education,
3. J. Kleinberg and E. Tardos, "Algorithm Design", Addison-Wesley, 2005.
4. Michael T. Goodrich and Roberto Tamassia, "Algorithm Design: Foundations, Analysis, and Internet Examples", Wiley.

Course Code:	**ITPC-204**
Subject Name	**OPERATING SYSTEM CONCEPTS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- To understand the services and design of an operating system.
- To understand the structure and organization of file system
- To understand the process states and various concepts such as scheduling and synchronization related with it.
- To understand different memory management approaches.
- Students should be able to use system calls for managing processes, memory and file system.
- To understand the data structures and algorithms for implementation of OS.

Course Outcomes

1. Understand functions, structures and history of operating systems
2. Able to know the design issues associated with operating systems
3. Master various process management concepts such as scheduling, synchronization, multithreading and deadlocks
4. Understand the various concepts associated with memory management such as virtual memory, demand paging, page replacements algorithms
5. Be familiar with various protection and security mechanisms

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-204												
CO 1	M	M	H	M	L							
CO 2					M	H						
CO 3	L	M		M								
CO 4								M				
CO 5							L					

Course Content

Introduction : Operating system and functions, Classification of Operating systems- Batch, Interactive, Time sharing, Real Time System, Multiprocessor Systems, Multiuser Systems, Multiprocess Systems, Multithreaded Systems, Operating System Structure- Layered structure, System Components, Operating System services, Reentrant Kernels, Monolithic and Microkernel Systems.

Concurrent Processes: Process Concept, Principle of Concurrency, Producer / Consumer Problem, Mutual Exclusion, Critical Section Problem, Dekker's solution, Peterson's solution, Semaphores, Test and Set operation; Classical Problem in Concurrency- Dining Philosopher Problem, Sleeping Barber Problem; Inter Process Communication models and Schemes, Process generation.

CPU Scheduling: Scheduling Concepts, Performance Criteria, Process States, Process Transition Diagram, Schedulers, Process Control Block (PCB), Process address space, Process identification information, Threads and their management, Scheduling Algorithms, Multiprocessor Scheduling. Deadlock: System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from deadlock.

Memory Management: Basic bare machine, Resident monitor, Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Protection schemes, Paging, Segmentation, Paged segmentation, Virtual memory concepts, Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing, Cache memory organization, Locality of reference.

I/O Management and Disk Scheduling: I/O devices, and I/O subsystems, I/O buffering, Disk storage and disk scheduling, RAID. File System: File concept, File organization and access mechanism, File directories, and File sharing, File system implementation issues, File system protection and security.

Recommended books:

1. Silberschatz, Galvin and Gagne, "Operating Systems Concepts", 8th Ed. Wiley, 2009.
2. Sibsankar Halder and Alex A Aravind, "Operating Systems", Pearson Education, 2009
3. Harvey M Dietel, " An Introduction to Operating System", 2nd Ed. Longman Higher Education.
4. D M Dhamdhere, "Operating Systems : A Concept based Approach", 3rd Ed., TMH, 2012.
5. William Stallings, "Operating Systems: Internals and Design Principles", 6th Ed., Pearson Education, 2009.

Course Code:	**ITPC-206**
Subject Name	**JAVA PROGRAMMING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objectives

- Building robust applications using Java's object-oriented features.
- Understanding the usage of java class libraries.
- Building multithreaded, platform-independent and GUI based java applications for business problems.

Course Outcomes

1. Write Java programs that solve simple business problems.
2. Create java applications that are robust and multithreaded.
3. Write simple GUI interfaces for a program to interact with users, and to understand the event-based GUI handling principles.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-206												
CO 1	M	M	H	M	L							
CO 2					M	H						
CO 3	L	M		M								

Course Content

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.

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.

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.

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes, inner classes. The AWT class hierarchy, user interface components- labels, button, canvas, scrollbars, text components, check box, check box groups, choices, lists panels – scroll pane, dialogs, menubar, graphics, layout manager – layout manager types – boarder, grid, flow, card and grib bag.

Applets: Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets.

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.

Networking: Basics of network programming, addresses, ports, sockets, simple client server program, multiple clients, Java .net package Packages – java.util,

Recommended books:

1. J.Nino and F.A. Hosch, “An Introduction to programming and OO design using Java”, 2nd Ed., John Wiley & sons.
2. T. Budd, “An Introduction to OOP”, 3rd Ed., Addison-Wesley
3. Y. Daniel Liang, “Introduction to Java programming”, 6th Ed., Prentice Hall, 2006.
4. R.A. Johnson-Thomson , “An introduction to Java programming and object oriented application development”, 1st Ed., Cengage Learning, 2006.
5. Cay.S.Horstmann and Gary Cornell, “Core Java 2, Vol 1, Fundamentals”, 5th Ed., Prentice Hall, 2000.

Course Code:	**ITPC-208**
Subject Name	**DATA MINING AND DATA WAREHOUSING CONCEPTS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Develop an understanding on the basic concepts, principles and applications of data mining and warehousing, also able to identify the scope and necessity of system.
- Describe the theoretical constructs and core processes of data mining and warehousing, also demonstrate the importance of paradigms from the fields of Artificial Intelligence and Machine

Learning to data mining.

- Comprehend contemporary issues related to research domain, industry and government, also able to inculcate professional, ethical issues and social responsibilities.

Course Outcomes

1. Develop an understanding on the basic concepts, principles and applications of data mining and warehousing, also able to identify the scope and necessity of system.
2. Describe the theoretical constructs and core processes of data mining and warehousing, also demonstrate the importance of paradigms from the fields of Artificial Intelligence and Machine Learning to data mining.
3. To understand the strengths and limitations of various data mining and warehousing models and techniques in solving the real world problems.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-208												
CO 1	M	M	H	M	L							
CO 2					M	H						
CO 3	L	M		M								

Course content

Introduction: Overview, Motivation(for Data Mining),Data Mining-Definition & Functionalities, Data Processing, Form of Data Pre-processing, Data Cleaning: Missing Values, Noisy Data,(Binning, Clustering, Regression, Computer and Human inspection),Inconsistent Data, Data Integration and Transformation. Data Reduction:-Data Cube Aggregation, Dimensionality reduction, Data Compression, Numerosity Reduction, Clustering, Discretization and Concept hierarchy generation

Concept Description: Definition, Data Generalization, Analytical Characterization, Analysis of attribute relevance, Mining Class comparisons, Statistical measures in large Databases. Measuring Central Tendency, Measuring Dispersion of Data, Graph Displays of Basic Statistical class Description, Mining Association Rules in Large Databases, Association rule mining, mining Single-Dimensional Boolean Association rules from Transactional Databases– Apriori Algorithm, Mining Multilevel Association rules from Transaction Databases and Mining Multi-Dimensional Association rules from Relational Databases

Classification and Predictions: What is Classification & Prediction, Issues regarding Classification and prediction, Decision tree, Bayesian Classification, Classification by Back propagation, Multilayer feed-forward Neural Network, Back propagation Algorithm, Classification methods K-nearest neighbour classifiers, Genetic Algorithm.

Cluster Analysis: Data types in cluster analysis, Categories of clustering methods, Partitioning methods.

Hierarchical Clustering- CURE and Chameleon, Density Based Methods-DBSCAN, OPTICS, Grid Based Methods- STING, CLIQUE, Model Based Method –Statistical Approach, Neural Network approach, Outlier Analysis

Data Warehousing: Overview, Definition, Delivery Process, Difference between Database System and Data Warehouse, Multi Dimensional Data Model, Data Cubes, Stars, Snow Flakes, Fact Constellations, Concept hierarchy, Process Architecture, 3 Tier Architecture, Data Marting.

Aggregation, Historical information, Query Facility, OLAP function and Tools. OLAP Servers, ROLAP, MOLAP, HOLAP, Data Mining interface, Security, Backup and Recovery, Tuning Data Warehouse, Testing Data Warehouse.

Recommended books:

1. M.H.Dunham,"Data Mining: Introductory and Advanced Topics", 1st Ed., Pearson Education, 2006.
2. Jiawei Han, Micheline Kamber, "Data Mining Concepts & Techniques", 3rd Ed., Elsevier, 2007.
3. Sam Anahory, Dennis Murray, "Data Warehousing in the Real World : A Practical Guide for Building Decision Support Systems", Pearson, 1997.
4. Mallach, "Data Warehousing System", McGraw –Hill, 2000.
5. Alex Berson, Stephen J. Smith, "Data Warehousing, Data Mining, & Olap", Tata McGraw-Hill Education, 2004.

Course Code:	**ITPC-210**
Subject Name	**FORMAL LANGUAGE AND AUTOMATA THEORY**
Contact hours/Credit Scheme (L-T-P-C)	**3-1-0-4**
Pre-requisites	**NONE**

Course Objectives

- To define mathematical methods of computing devices, called abstract machines, namely Finite Automata, Pushdown Automata, and Turning Machines.
- To study the capabilities of these abstract machines.
- To classify machines by their power to recognize languages.
- Employ finite state machines to solve problems in computing
- Explain deterministic and non- deterministic machines.
- Identify different formal language classes and their relationships
- Design grammars and recognizers for different formal languages
- Determine the decidability and intractability of computational problems
- Comprehend the hierarchy of problems arising in the computer sciences

Course Outcomes

1. To explain basic concepts in formal language theory, grammars, automata theory, computability theory, and complexity theory.
2. To demonstrate abstract models of computing, including deterministic (DFA), non-deterministic (NFA), Push Down Automata (PDA) and Turing (TM) machine models and their power to recognize the languages.
3. To explain the application of machine models and descriptors to compiler theory and parsing.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-210												
CO 1	M	M	H	M	L							
CO 2					M	H						
CO 3	L	M		M								

Course Content

Introduction: Alphabets, Strings and Languages; Automata and Grammars, Deterministic finite Automata (DFA)-Formal Definition, Simplified notation: State transition graph, Transition table, Language of DFA, Nondeterministic finite Automata (NFA), NFA with epsilon transition, Language of NFA, Equivalence of NFA and DFA, Minimization of Finite Automata, Distinguishing one string from other, Myhill-Nerode Theorem

Regular expression (RE), Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleen’s Theorem, Regular expression to FA, DFA to Regular expression, Arden Theorem, Non Regular Languages, Pumping Lemma for regular Languages . Application of Pumping Lemma, Closure properties of Regular Languages, Decision properties of Regular Languages, FA with output: Moore and Mealy machine, Equivalence of Moore and Mealy Machine, Applications and Limitation of FA.

Context free grammar (CFG) and Context Free Languages (CFL): Definition, Examples, Derivation , Derivation trees, Ambiguity in Grammar, Inherent ambiguity, Ambiguous to Unambiguous CFG, Useless symbols, Simplification of cfgs, Normal forms for cfgs: CNF and GNF, Closure proper ties of cfls, Decision Properties of cfls: Emptiness, Finiteness and Membership, Pumping lemma for cfls,

Push Down Automata (PDA): Description and definition, Instantaneous Description, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic PDA, Equivalence of PDA and CFG, CFG to PDA and PDA to CFG, Two stack PDA

Turing machines (TM): Basic model, definition and representation, Instantaneous Description, Language acceptance by TM, Variants of Turing Machine, TM as Computer of Integer functions, Universal TM, Church’s Thesis, Recursive and recursively enumerable languages, Halting problem, Introduction to Undecidability, Undecidable problems about tms. Post correspondence problem (PCP), Modified PCP, Introduction to recursive function theory

Recommended books:

1. Hopcroft, Ullman, “Introduction to Automata Theory, Languages and Computation”, Pearson Education
2. K.L.P. Mishra and N.Chandrasekaran, “Theory of Computer Science : Automata, Languages and Computation”, PHI
3. Martin J. C., “Introduction to Languages and Theory of Computations”, TMH
4. Papadimitrou, C. and Lewis, C.L., “Elements of the Theory of Computation”, PHI
5. John E. Hopcroft, “Automata Theory Language & Computation”, Prentice Hall; 3 edition.

Course Code:	**HMCI-201**
Subject Name	**ECONOMICS FOR ENGINEERs**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

COURSE ASSESSMENT METHODS: Two sessional exams and one end-semester exam, along with assignments, presentations and class tests which may be conducted by the course coordinator in lieu of internal assessment.

Course Objectives

1. To gain knowledge about different micro and macro aspects of economics.
2. To understand the complex relationships and intricacies of different economic variables.
3. To understand the micro and macro implications of economic policies and decisions.

Course Outcomes

The students will able to understand different terms and concepts of economics. The students will gain proficiency in understanding the changes in economic environment and their impact both at micro and macro levels.

Course Contents

Basic Economic concepts, Decision making under risk and uncertainty. Concept of utility, demand and supply, elasticity of demand and supply, Demand forecasting. Production function in short and long run: law of diminishing marginal returns, isoquant-isocost approach. Economies of scale. Shapes of different cost curves in short and long run. Price-output determination in perfect competition, monopoly, monopolistic competition and oligopoly. Macroeconomics: national income, business cycle, fiscal policy, monetary policy, price indices, inflation, theories of international trade.

Reference Books

1. Carl E Case, Ray C Fair and Sharon E Oster (2017), Principles of Economics, Pearson
2. John Sloman, Dean Garratt and Alison Wride (2014), Economics, 9th edition, Pearson.
3. Christopher R Thomas, S Charles Maurice and Sumit Sarkar (2010), Managerial Economics, 9th edition, McGraw Hill Publication.
4. H L Ahuja (2017), Managerial Economics, 9th edition, S Chand Publishing.

Course Code:	**ITPC-222**
Subject Name	**INTRODUCTION TO DESIGN & ANALYSIS OF ALGORITHMS LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives:

- To introduce mathematical aspects and analysis of algorithms.
- To introduce algorithm design methods

Course Outcomes

1. Identify the problem given and design the algorithm using various algorithm design techniques.
2. Implement various algorithms in a high level language.
3. Analyze the performance of various algorithms.
4. Compare the performance of different algorithms for same problem

LIST OF PRACTICALS

1. Implementation of Quick sort using the Divide and Conquer technique and analyze its Time Complexity.
2. Implementation of Merge sort using the Divide and Conquer technique and analyze its Time Complexity.
3. Implementation of Radix sort and analyze its Time Complexity.
4. Implementation of Counting sort and analyze its Time Complexity.
5. Implementation of Bucket sort and analyze its Time Complexity.
6. Implementation of Hash table and supported operations.
7. Implementation of red black trees and supported operations.
8. Implementation of Max/Min Heap and supported operations
9. Implementation of the Minimum Spanning tree using prim's algorithm and kruskal's algorithm.
10. Implementation of matrix multiplication problem using the dynamic programming approach.
11. Implementation of assembly line scheduling using the dynamic programming approach.
12. Implementation of the longest common subsequence problem using the dynamic programming approach.
13. Implementation of the Optimal Binary Search Tee problem using the dynamic programming approach.
14. Implementation of the knapsack problem using greedy method.
15. Implementation to find the solution to the N queen's problem using backtracking.
16. Implementation to find the shortest path using Floyd's algorithm.
17. Implementation to solve Graph Coloring problem and Hamiltonian Cycle Problem.

This is only the suggested list of Practicals. Instructor may frame the Practicals relevant to the course contents.

Course Code:	**ITPC-224**
Subject Name	**OPERATING SYSTEMS CONCEPTS LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

- Analyse the working of an operating system and its components and Define and analyse the synchronization process.
- Identify the working methodology of multithreaded applications.
- Compare and analyse different file systems being used in different operating systems.
- Design and implement system-level applications for open-source operating systems

Course outcomes

1. Analyse the working of an operating system and its components.
2. Define and analyse the synchronization process.
3. Identify the working methodology of multithreaded applications.
4. Compare and analyse different file systems being used in different operating systems.

LIST OF PRACTICALS

1. Simulation of the CPU scheduling algorithms
   a) Round Robin
   b) SJF
   c) FCFS
   d) Priority
2. Simulation of MUTEX and SEMAPHORES.
3. Simulation of Bankers Deadlock Avoidance and Prevention algorithms.
4. Implementation of Process Synchronization (Reader-Writer, Sleeping Barber and Dining Philosopher's Problem)
5. Simulation of page Replacement Algorithms
   a) FIFO
   b) LRU
   c) LFU
6. Simulation of paging techniques of memory management.
7. Simulation of file allocation Strategies
   a) Sequential
   b) Indexed
   c) Linked
8. Simulation of file organization techniques
   a) Single Level Directory
   b) Two Level
   c) Hierarchical
   d) DAG

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code:	**ITPC-226**
Subject Name	**JAVA PROGRAMMING LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

- To make the student learn the application of advanced object oriented concepts for solving problems.
- To teach the student to write programs using advanced Java features to solve the problems

Course outcomes

1. knowledge of the structure and model of the Java programming language
2. use the Java programming language for various programming technologies
3. evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements

LIST OF PRACTICALS

1. Program to find total, average of given two numbers by using function with default arguments, static data members and this keyword.
2. Program to illustrate class and objects.
3. Program to illustrate constructors.
4. Program to create a class complex with necessary operator overloading and type conversion such as integer to complex, complex to double.
5. Program that randomly generates complex numbers and write two numbers per line in a file along with an operator(+,-,P,*,/) .The numbers are written to file in the format (a+ib)
6. Program to read one line at a time, perform the corresponding operation on two
7. complex numbers read, write the result to another file (one per line)
8. Program to illustrate inheritance (Student Evaluation)
9. Program to handle the situation of exception handling.
10. Program to demonstrate the concept of polymorphism.
11. Program to illustrate Method Overriding?
12. Program to illustrate Method overloading of assignment operator?
13. Program to illustrate Array Manipulation?
14. Program to illustrate Synchronization?
15. Program to StringTokenizer?
16. Program to implement the concept of User defined Exceptions.
17. Program to illustrate the use of creation of packages.
18. Program to illustrate Multithreading and Multitasking?
19. Program to illustrate thread priorities.
20. Program to illustrate applet concept.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code:	**ITPC-228**
Subject Name	**Data Mining and Data Warehousing Concepts Lab**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

- To understand the need of Data Mining and advantages to the business and scientific world.

Course outcomes

1. Synthesize the data mining fundamental concepts and techniques from multiple perspectives.
2. Develop skills and apply data mining tools for solving practical problems
3. Develop research skills by reading the data mining literature and develop advance relevant programming skills

LIST OF PRACTICALS

Students are required to perform practical’s in Oracle/MS SQL Server and STATISTICA *Data Miner*

1. Building a Database Design using ER Modeling and Normalization Techniques
2. Implementation of functions ,Procedures, Triggers and Cursors
3. Load Data from heterogeneous sources including text files into a predefined warehouse schema.
4. Design a data mart for a bank to store the credit history of customers in a bank. Use this credit profiling to process future loan applications.
5. Perform feature Selection and Variable Filtering (for very large data sets)
6. Perform association Mining in large data sets using FP, Apriori and ECLAT algorithm
7. Execute various data cluster operations like Drill-Down, Roll up, Slice and Dice operations
8. Implementation of generalized EM & *k*-Means Cluster Analysis.
9. Implementation of generalized Additive Models (GAM)
10. Implement decision tree algorithm like ID3, CART, and HUNT.
11. Implementation of general CHAID (Chi-square Automatic Interaction Detection) Models
12. To perform “Goodness of Fit” Computations using various techniques.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Fifth Semester

S.No.	Sub. Code	Subjects	L	T	P	Credits	
1	ITPC-301	Cryptography and Network Security	3	0	0	3	PC
2	ITPC-303	Software Engineering Concepts	3	0	0	3	PC
3	ITPC-305	Web Design Technologies	3	0	0	3	PC
4	ITPC-307	Computer Graphics and Applications	3	0	0	3	PC
5	ITPC-309	Discrete Mathematics	3	1	0	4	PC
6	ITPE-XXX	DE –I	3	0	0	3	PE
Practical /Training /Projects							
7	ITPC-321	Cryptography and Network Security Lab	0	0	2	1	PC
8	ITPC-323	Software Engineering Concepts Lab	0	0	2	1	PC
9	ITPC-325	Web Design Technologies Lab	0	0	2	1	PC
10	ITPC-327	Computer Graphics and Applications Lab	0	0	2	1	PC
11	ITCI-300	Minor Project Phase-I	0	0	2	0*	CIC
TOTAL			**18**	**1**	**10**	**23**	

Course Code:	**ITPC-301**
Subject Name	**CRYPTOGRAPHY AND NETWORK SECURITY**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- To understand Mathematical foundations of Cryptography theory.
- To analyze the robustness of Cryptosystems.
- To design robust cryptosystem for real time applications.
- To develop prototype Cryptosystems and demonstrate their performance.

Course Outcomes

1. To understand Mathematical foundations of Cryptography theory.
2. To analyze the robustness of Cryptosystems.
3. To design robust cryptosystem for real time applications.
4. To develop prototype Cryptosystems and demonstrate their performance.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-301												
CO 1	M		M			H		M		M		L
CO 2	M	H	M	M		H		H				H
CO 3			H	M	H	M		H	H			M
CO 4				M	M				M		M	

Course Content

Introduction to security attacks, services and mechanism, classical encryption techniques- substitution ciphers and transposition ciphers, cryptanalysis, steganography, Stream and block ciphers.

Modern block ciphers: block ciphers principles, Shannon's theory of confusion and diffusion, fiestal structure.

Introduction to group, field, finite field of the form GF(p), modular arithmetic, prime and relative prime numbers, Extended Euclidean Algorithm, Data Encryption Standard(DES), strength of DES, idea of differential cryptanalysis, block cipher modes of operations, Triple DES, Advanced Encryption Standard (AES) encryption and decryption Fermat's and Euler's theorem, primality testing, Chinese remainder theorem, discrete logarithmic problem, principles of public key crypto systems, RSA algorithm, security of RSA

Message authentication codes: authentication requirements, authentication functions, message authentication code, hash functions, birthday attacks, security of hash functions, Secure Hash Algorithm (SHA)

Digital signatures: digital signatures, elgamal digital signature techniques, Digital Signature Standards (DSS), proof of digital signature algorithm,

Key management and distribution: symmetric key distribution, Diffie -Hellman key Exchange, Public key distribution, X.509 Certificates, Public key Infrastructure.

Authentication applications: Kerberos

Electronic mail security: Pretty Good Privacy (PGP), S/MIME.

IP security: architecture, authentication header, encapsulating security payloads, combining security associations, key management.

Introduction to Secure Socket Layer, Secure Electronic Transaction (SET)

System security: introductory idea of intrusion, intrusion detection, viruses and related threats, firewalls

Recommended books:

1. William Stallings, “Cryptography and network Security”, Pearson Education 2003.
2. Trappe & Washington, “Introduction to Cryptography with Coding Theory”, Prentice-Hall 2001
3. D Stinson, “Cryptography: Theory and Practice”, Second Edition Chapman & Hall 2002.
4. Kaufman, Perlman, and Speciner, “Network Security”, Prentice-Hall Second Edition 2001.
5. Michael E. Whitman, “Principles of information Security” , Cengage learning, New Delhi

Course Code:	**ITPC-303**
Subject Name	**SOFTWARE ENGINEERING CONCEPTS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Study the current software engineering techniques and examines the software life-cycle, including software specification, design implementation, testing and maintenance.
- Present software engineering methodologies for the development of Quality, cost-effective, schedule-meeting software.
- Develop an understanding of ethical and professional issues related to Software Project Delivery.

Course Outcomes

1. Able to apply the concepts and choose an appropriate SDLC process model for user requirements.
2. To analyze requirement techniques like Data flow diagram, Entity relationship diagram etc.
3. Understanding the concept of Software Design and emphasizing upon various software metrics used for analyzing the software.
4. Demonstrate various testing methodologies and debugging tools for a prototype software.
5. Design various software reliability measures to assess the quality of software in case of various faults and failures.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-303												
CO 1.	M	M				L						
CO 2.		H			L					L		
CO 3.	H		H		M							
CO 4.			L	H	M				M			
CO 5.		M	M		L							

Course content

Introduction to Software Engineering: What is Software Engineering?, Why Software Engineering, Software Crisis, Notable Changes in Software Development Practices, Software myths.

The Software Process: Plan-driven and agile processes, different development philosophies: sequential vs iterative, software development life cycle (SDLC), overview of various SDLC models/methodologies. Plan-Driven Development: The Waterfall Model, Incremental development, Integration and configuration, How to Cope with changes?, Software prototyping vs Incremental delivery.

Agile-Software Development: Agility, Extreme Programming, Agile Project Management (Scrum), Scaling out and scaling up, Problem with Agile methods, Combining Agile and plan-driven methods, Agile methods across organizations Process models: The waterfall model, Incremental process models, Evolutionary process models, The Unified process.

Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

Requirements Engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

Software Design Concepts: Activities carried out during design, Classification of Design Methodologies: Function-Oriented, Object-Oriented, Aspect-oriented, Component-based, Properties of a Good Design, Layered Design, Modularity

Function Oriented Design: Overview of SA/SD Methodology, Data Flow Diagrams (DFDs), DFD Model of a System, DFD Model to Structure Chart

Object-Oriented Design: Objects and object classes, An Object-Oriented design process, UML.

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.

Risk management: Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM.

Quality Management: Quality concepts, Software quality assurance, Software reliability, The ISO 9000 quality standards.

Recommended books:

1. Fundamentals of Software Engineering, Rajib Mall, Fourth Edition, PHI, 2016.
2. Software Engineering- Sommerville, 7th edition, Pearson education.
3. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
4. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition. McGrawHill International Edition.

Course Code:	**ITPC-305**
Subject Name	**WEB DESIGN TECHNOLOGIES**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Introduction and brief history of world wide web (WWW).
- Web essentials: HTML, XHTML, CSS.
- Addressing web standards, audience requirements and principles of web page design.
- Introduction of Web architecture, databases, jdbc

Course Outcomes

1. Understand basic principles of web site design, considering the information architecture.
2. Incorporate best practices in navigation, usability in website design
3. Design of website adhering to current web standards (HTML, XML, CSS)
4. Learning various scripting languages to create interactive components in web pages.

Course Outcomes	**Program outcomes**											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITPC-305												
CO 1	H		M		M							
CO 2	M	M	H	M	M				H			
CO 3					H	M						
CO 4		H		H		H		H		M		M

Course Content

Introduction to HTML: HTML Common tags- List, Tables, images, forms, Frames; Cascading Style sheets;

Introduction to JavaScript: Scripts, Objects in Java Script, Dynamic HTML with Java Script

XML: Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX

Java Beans: Introduction to Java Beans, Advantages of Java Beans, BDK Introspection, Using Bound properties, Bean Info Interface, Constrained properties Persistence, Customizes, Java Beans API, Introduction to EJB's

Web Servers and Servlets: Tomcat web server, Introduction to Servelets: Lifecycle of a Serverlet, JSDK, The Servelet API, The javax.servelet Package, Reading Servelet parameters, Reading Initialization parameters. The javax.servelet HTTP package, Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues,

Introduction to JSP: The Problem with Servelet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC Setting Up and JSP Environment: Installing the Java Software Development Kit, Tomcat Server & Testing Tomcat

JSP Application Development: Generating Dynamic Content, Using Scripting Elements Implicit JSP Objects, Conditional Processing – Displaying Values Using an Expression to Set an Attribute, Declaring Variables and Methods Error Handling and Debugging Sharing Data Between JSP pages, Requests, and Users Passing Control and Date between Pages – Sharing Session and Application Data – Memory Usage Considerations

Database Access : Database Programming using JDBC, Studying Javax.sql.* package, Accessing a Database from a JSP Page, Application – Specific Database Actions, Deploying JAVA Beans in a JSP Page, Introduction to struts framework.

Recommended books:

1. Chris Bates ,"Web Programming, building internet applications", 3rd Ed., WILEY Dreamtech, 2007.
2. Patrick Naughton and Herbert Schildt, "The complete Reference Java 2" 5th Ed., TMH, 2002.
3. Hans Bergsten , "java Server Pages", 2nd Ed., O'Reilly, 2002.
4. Dietel, Nieto, "Internet and World Wide Web – How to program", 2nd Ed., PHI, 2001.
5. Joel Sklar, "Web Warrier guide to web design technologies", 3rd Ed. Cengage Learning, 2005.

Course Code:	**ITPC-307**
Subject Name	**Computer Graphics and Applications**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Comprehensive introduction for computer graphics leading to the ability to understand contemporary terminology, progress, issues, and trends
- Involves processing and transforming large, incomprehensible datasets into a format more easily understood by humans
- Introduce theoretical methods for two-dimensional and three-dimensional graphics with applications to visualization techniques

Course Outcomes

1. Use the underlying architecture, algorithms, mathematical concepts, supporting computer animation.
2. Enhance their perspective of modern computer system with modeling, analysis and interpretation of 2D and 3D
3. Fundamental graphic design theory and principles as they relate to current trends in visual

communication

4. Organize information to make compelling and experimental visual expressions for presentation

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITPC-307												
CO 1		M			L							
CO 2	L		M					M				
CO 3					M				L			
CO 4		L										

Course Content

Introduction: Introduction, Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations and input devices

Output primitives: Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms

2-D geometrical transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems.

2-D viewing: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm.

3-D object representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces. Basic illumination models, polygon rendering methods.

3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations.

3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

Visible surface detection methods: Classification, back-face detection, depth-buffer, scan-line, depth sorting, BSP-tree methods, area sub-division and octree methods.

Computer animation: Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications.

Recommended books:

1. Edward Angel and Dave Shreiner, Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL, Pearson, 6th edition, 2011.
2. F.S. Hill Jr., Computer Graphics Using OpenGL, Pearson, 3rd edition, 2006.
3. John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, and Kurt Akeley, Computer Graphics: Principles and Practice, Pearson Education, 1st edition, 2013.

Course Code:	**ITPC-309**
Subject Name	**DISCRETE MATHEMATICS**
Contact hours/Credit Scheme (L-T-P-C)	**3-1-0-4**
Pre-requisites	**NONE**

Course Objectives

- To learn basic logic and set theory
- To learn core ideas in combinatorial mathematics
- To learn core ideas in graph theory

Course Outcomes

1. To introduce a number of Discrete Mathematical Structures (DMS) found to be serving as tools even today in the development of theoretical computer science.
2. Course focuses on of how Discrete Structures actually helped computer engineers to solve problems occurred in the development of programming languages.
3. Also, course highlights the importance of discrete structures towards simulation of a problem in computer science and engineering.
4. Introduction of a number of case studies involving problems of Computer Technology.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-309												
CO 1	H		M									
CO 2	M	H			M							
CO 3		M	M		H							
CO 4		M	H	M	M	M						

Topics Covered

Set Theory: Definition of sets, countable and uncountable sets, Venn Diagrams, proofs of some general identities on sets Set Theory, Functions and Relations: Subsets, Power Set, Null Set, Singleton, Finite Set, Infinite Set, Universal Set, Disjoint Sets, Operation on Sets, Venn Diagrams, Cartesian Product of Sets, Partition of Sets, Concept of Relation & Properties of Relations, Different types of Relations, Tabular and Matrix Representation of Relations, Relations and Diagraphs, Composition of Relations, Functions and their different mappings, Composition of Function, Recursion and Recurrence Relations.

Algebraic Structures: Definition, Properties, types: Semi Groups, Monoid, Groups, Abelian group, properties of groups, Subgroup, cyclic groups, Cosets, factor group, Permutation groups, Normal subgroup, Homomorphism and isomorphism of Groups, example and standard results, Rings and Fields: definition and standard results.

Posets, Hasse Diagram and Lattices: Introduction, ordered set, Hasse diagram of partially, ordered set, isomorphic ordered set, well ordered set, properties of Lattices, bounded I and complemented lattices.

Boolean Algebra: Partial Ordering, Totally ordered Sets, Dual Order, Hasse Diagram Lexicographic Ordering, Cover of an Element, Least and Greatest Elements, Minimal and

Maximal Elements, Upper and Lower Bound, Well-Order Set, Binary and n-Ary Operations, Lattices, Atoms of a Boolean Algebra, Boolean Expressions, Applications of Boolean Algebra to Switching Theory.

Tree: Definition, Rooted tree, properties of trees, binary search tree, tree traversal.

Propositional Logic: Proposition, First order logic, Basic logical operation, truth tables, tautologies, Contradictions, Algebra of Proposition, logical implications, logical equivalence, predicates, Universal and existential quantifiers.

Combinatorics & Graphs: Recurrence Relation, Generating function., Simple graph, multi graph, graph terminology, representation of graphs, Bipartite, Regular, Planar and connected graphs, connected components in a graph, Euler graphs, Hamiltonian path and circuits, Graph coloring, chromatic number, isomorphism and Homomorphism of graphs.

Recommended books

1. Liptschutz, Seymour, “ Discrete Mathematics”, McGraw Hill.
2. Trembley, J.P & R. Manohar, “Discrete Mathematical Structure with Application to Computer Science”, McGraw Hill.
3. Kenneth H. Rosen, “Discrete Mathematics and its applications”, McGraw Hill.
4. Deo, Narsingh, “Graph Theory With application to Engineering and Computer.Science.”, PHI.
5. Krishnamurthy, V., “Combinatorics Theory & Application”, East-West Press Pvt. Ltd., New Delhi.

Course Code:	**ITPC-321**
SUBJECT NAME	**CRYPTOGRAPHY AND NETWORK SECURITY LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

- Understand cryptography and network security concepts and application
- Identify and investigate network security threat
- Analyze and design network security protocols

Course outcomes

1. Identify Vulnerabilities in a Network
2. Solve Problems using various Algorithms
3. Understand Web And DNS Security

LIST OF PRACTICALS

1. Implement the encryption and decryption of 8-bit data using ‘Simplified DES Algorithm’ in ‘C’.
2. Implement ‘Linear Congruential Algorithm’ to generate 5 pseudo-random numbers in ‘C’.
3. Implement Rabin-Miller Primality Testing Algorithm in ‘C’.
4. Implement the Euclid Algorithm to generate the GCD of an array of 10 integers in ‘C’.
5. Implement RSA algorithm for encryption and decryption in ‘C’.
6. Configure a mail agent to support Digital Certificates, send a mail and verify the correctness of this system using the configured parameters.
7. Configure SSH (Secure Shell) and send/receive a file on this connection to verify the correctness of this system using the configured parameters.
8. Configure a firewall to block the following for 5 minutes and verify the correctness of this system using the configured parameters:
   a. Two neighborhood IP addresses on your LAN
   b. All ICMP requests
   c. All TCP SYN Packets

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code:	**ITPC-323**
Subject Name	**SOFTWARE ENGINEERING CONCEPTS LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

- To provide the students with an overview of different software development life cycle models.
- To enable the students to apply a model as per scenario given to them.
- Identify the important parts of an SRS document; identify functional and non-functional requirements for a project.
- Identify the important items that should be designed during high-level and detailed design activities.

Course Outcomes

1. Plan a software engineering process life cycle
2. Able to elicit, analyze and specify software requirements.
3. Analyze and translate a specification into a design.
4. Realize design practically, using an appropriate software engineering methodology.
5. Able to use modern engineering tools for specification, design, implementation, and testing

LIST OF PRACTICALS

1. Introduction and project definition.
2. Introduction to Project Management Software.
3. Software process overview.
4. Project planning.
5. Software Requirements Specification (SRS).
6. Introduction to UML and use case diagrams.
7. Software Design: software architecture and object-oriented design/function oriented design.
8. Effort and cost estimation using COCOMO
9. Issue Tracking Systems like GitHub, BitBucket.
10. Software Configuration Management tools like CVS or SVN.
11. Software testing.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code:	**ITPC-325**
Subject Name	**WEB DESIGN TECHNOLOGIES LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

- To develop an ability to design and implement static and dynamic websites.
- Apply markup languages to design effective web pages.
- Design the DB applications using JDBC, JSP and servlets.
- Design dynamic and interactive web pages by embedding Java Script code in HTML.
- Use Java Script to validate user input and create good, effective and customized websites

Course Outcomes

1. Analyze a web page and identify its elements and attributes.
2. Create web pages using XHTML and Cascading Style Sheets.

3. Build dynamic web pages using JavaScript (Client side programming).
4. Create XML documents and Schemas.

LIST OF PRACTICALS

1. Program using HTML/Java scripts to display your CV in Web Browser.
2. Creation and annotation of static web pages using any HTML editor.
3. Create a web page which includes a map and display the related information when a hot spot is clicked in the map.
4. Create the several Frames using HTML and display to the web browser.
5. Create Scientific Calculator using JavaScript.
6. To design web page to create a real time clock with a timing event using java script event handling mechanism.
7. Create a web page that displays college information using Style Sheet.
8. Create a Client Side Scripts for Validating Web Form Controls using DHTML.
9. Program to use XML and JavaScript for creation of your homepage.
10. Program in XML for creation of DTD which specifies a particular set of rules.
11. Create a Stylesheet in CSS/XSL and display the document in Web Browser.
12. Implement a Java Servelet for HTTP Proxy Server.
13. Use JSP pages for sharing session and application data of HTTP Server.
14. Program to use JDBC connectivity program for maintaining database by sending queries.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code:	**ITPC-327**
Subject Name	**COMPUTER GRAPHICS AND APPLICATIONS LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

- To discover the fundamental principles of animation & to exploit these fundamental principles to create convincing character motion.

Course Outcomes

1. Understand the basic concepts of computer graphics
2. Apply clipping and filling techniques for modifying an object.
3. Understand the concepts of different type of geometric transformation of objects in 2D and 3D.
4. Understand the practical implementation of modeling, rendering, viewing of objects in 2D

List of Practicals

1. Implementation of a line using DDA Algorithm.
2. Implementation of a line using Bresenham’s Algorithm.
3. Implementation of a circle using trigonometric Algorithm.
4. Implementation of a circle using Bresenham’s Algorithm.
5. Implementation of a circle using Midpoint Algorithm.
6. Implementation of an ellipse using Trigonometric Algorithm.
7. Implementation of an ellipse using Midpoint Algorithm.
8. To translate an object with translation parameters in X and Y directions.
9. To scale an object with scaling factors along X and Y directions.

10. To rotate an object with a certain angle.
11. To perform composite transformations of an object.
12. To clip line segments against windows.
13. Demonstrate the properties of Bezier Curve.
14. Run a sample session on Microsoft Windows including the use of Paintbrush.
15. Implementation of simple graphics animation.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

SIXTH SEMESTER

S.No.	Sub. Code	Subjects	L	T	P	Credits	
1	ITPC-302	Soft Computing Concepts	3	0	0	3	PC
2	ITPC-304	Object-Oriented Modeling and Design with UML	3	0	0	3	PC
3	ITPC-306	Mobile Application Development	3	0	0	3	PC
4	ITPC-308	Machine Learning Concepts	3	0	0	3	PC
5	ITPE-XXX	DE-II	3	0	0	3	PE
6	ITOE-XXX	OE -I	3	0	0	3	OE
Practical /Training /Projects							
7	ITPC-322	Soft Computing Concepts Lab	0	0	2	1	PC
8	ITPC-324	Object-Oriented Modeling and Design with UML Lab	0	0	2	1	PC
9	ITPC-326	Mobile Application Development Lab	0	0	2	1	PC
10	ITCI-301	Minor Project Phase-II	0	0	2	2*	CIC
TOTAL			**18**	**0**	**8**	**23**	

Course Code:	**ITPC-302**
Subject Name	**SOFT COMPUTING CONCEPTS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Motivation and historical background of Soft Computing.
- Application of Fuzzy logic.
- Biologically inspired algorithm such as neural networks, genetic algorithms, ant colony optimization, and bee colony optimization.
- Hybrid systems of neural network, genetic algorithms and fuzzy systems.

Course Outcomes

1. Understand and describe soft computing techniques and their roles in building intelligent machines
2. Recognize the feasibility of applying a soft computing methodology for a particular problem
3. Effectively use existing software tools to solve real problems using a soft computing approach
4. Evaluate and compare solutions by various soft computing approaches for a given problem.

Course Outcomes	**Program outcomes**											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITPC-302												
CO 1		M			M							
CO 2			M				L					
CO 3			L	L								
CO 4					M							

Course Content

Artificial Neural Networks: Basic concepts - Single layer perception - Multilayer Perception - Supervised and Unsupervised learning – Back propagation networks - Kohnen'sself organizing networks - Hopfield network.

Fuzzy Systems: Fuzzy sets and Fuzzy reasoning - Fuzzy matrices - Fuzzy functions - Decomposition - Fuzzy automata and languages - Fuzzy control methods - Fuzzy decision making.

Neuro - Fuzzy Modeling: Adaptive networks based Fuzzy interface systems - Classification and Regression Trees -Data clustering algorithms - Rule based structure identification - Neuro-Fuzzy controls -Simulated annealing – Evolutionary computation.

Genetic Algorithms: Survival of the Fittest - Fitness Computations - Cross over - Mutation - Reproduction -Rank method - Rank space method.

SoftcomputingAnd Conventional AI: AI search algorithm - Predicate calculus - Rules of interference – Semantic networks -Frames - Objects - Hybrid models - Applications.

Recommended books:

1. Jang J.S.R., Sun C.T. and Mizutani E, "Neuro-Fuzzy and Soft computing", Prentice Hall 1998.
2. LaureneFausett, "Fundamentals of Neural Networks", Prentice Hall, 1994.
3. George J. Klir and Bo Yuan, "Fuzzy sets and Fuzzy Logic", Prentice Hall, USA 1995.
4. N. J. Nelsson, "Artificial Intelligence - A New Synthesis", Harcourt Asia Ltd., 1998.
5. D.E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Addison Wesley, N.Y, 1989.

Course Code:	**ITPC-304**
Subject Name	**OBJECT-ORIENTED MODELING AND DESIGN WITH UML**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Understanding basic principles modeling of software systems using UML 2.0.
- Theoretical and practical preparation enabling students to work in project teams.

Course Outcomes

1. Explain the object- oriented software development process, including object-oriented methodologies and work flow.
2. Collect requirements and prepare their scenarios
3. Justify designs based on design principles, patterns, and heuristics.
4. Prepare diagrams by UML.

Course Outcomes	**Program outcomes**											
ITPC-304	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	H	M	M	L								
CO 2	L			H	H							
CO 3						M		M				
CO 4					L					H		

Course Outcomes

Introduction: What is Object Orientation? What is OO development? OO themes, OO modeling history, Modeling as Design Technique: Modeling, abstraction, The three models.

Process Overview: Development stages, Development life cycle.

Project Organization Concepts: Project Organizations, Roles, Tasks and Work Products, Schedule
Project Communication Concepts: Planned Communication, Unplanned Communication
Communication Mechanisms, Organizational Activities

Analysis: Problem Analysis, Problem Domain Classes, Defining the problem and the scope, Requirements Engineering, Types of Requirements. Requirements Validation, Completeness, Consistency, Clarity, and Correctness, Realism, Verifiability, and Traceability, Greenfield Engineering, Reengineering, and Interface Engineering

System Conception: Devising a system concept, elaborating a concept, preparing a problem statement, domain Analysis: Overview of analysis.

Class Modeling: Object and class concepts, Link and associations concepts, Generalization and inheritance, A sample class model, Navigation of class models, Advanced object and class concepts, Aggregation, Abstract classes, Packages.

State Modeling: Events, States, Transitions and Conditions, State diagrams, State diagram behavior, relation of class and state models

Interaction Modeling: Use case models, Sequence models, Activity models. Use case relationships.

Patterns: What is a pattern and what makes a pattern? Properties of Patterns, Pattern – A Three-part Schema, Different Types of Patterns: Layer Pattern, Broker Pattern, Shared-Data Pattern, Pipe and Filter

Pattern, Model-View-Controller Pattern, Client-Server Pattern, Peer-to-Peer Pattern, Microkernel Pattern, Blackboard, Relationship between tactics and patterns, using tactics together.

Recommended Books:

1. Bernd Bruegge & Allen H. Dutoit, "Object-Oriented Software Engineering Using UML, Patterns, and Java" Third Edition, Pearson.
2. Len Bass, Paul Clements, Rick Kazman, Software Architecture in Practice, Pearson Education, Third edition, 2013.
3. Michael Blaha, James Rumbaugh, "Object Oriented Modeling and Design with UML", Second Edition, PHI.
4. Meilir Page-Jones, "Fundamentals of Object Oriented Design in UML", Pearson Education (2008).
5. Simon Benett, Steve Mc Robb, "Object Oriented Systems Analysis and Design using UML", Second Edition, TMH (2007).

Course Code:	**ITPC-306**
Subject Name	**MOBILE APPLICATION DEVELOPMENT**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Introduction and characteristics of mobile applications.
- Application models of mobile application frameworks.
- Managing application data and User-interface design for mobile applications.
- Integrating networking, the OS and hardware into mobile-applications.
- Addressing enterprise requirements in mobile applications – performance, scalability, modifiability, availability and security.

Course Outcomes

1. Understand technology and business trends impacting mobile applications
2. Be competent with the characterization and architecture of mobile applications.
3. Understand enterprise scale requirements of mobile applications.
4. Design and develop mobile applications using one application development framework.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-306												
CO 1			M				L					
CO 2		L				M						
CO 3				M				H	H			
CO 4		L					L					

Course content

Introduction: Mobile operating system, Operating system structure, Constraints and Restrictions, Hardware configuration with mobile operating system, Features: Multitasking Scheduling, Memory Allocation, FileSystem Interface, Keypad Interface, I/O Interface, Protection and Security, Multimedia features.

J2ME Overview: Java 2 Micro Edition and the World of Java, Inside J2ME, J2ME and Wireless Devices Small Computing Technology: Wireless Technology, Radio Data Networks, Microwave Technology, Mobile Radio Networks, Messaging, Personal Digital Assistants.

J2ME Architecture and Development Environment: J2ME Architecture, Small Computing Device

Requirements, Run-Time Environment, MIDlet Programming, Java Language for J2ME, J2ME Software Development Kits, Hello World J2MEStyle, Multiple MIDlets in a MIDlet Suite, J2ME Wireless Toolkit

J2ME Best Practices and Patterns: The Reality of Working in a J2ME World, Best Practices. Commands, Items, and Event Processing: J2ME User Interfaces, Display Class, The Palm OS Emulator, Command Class, Item Class, Exception Handling

High-Level Display: Screens: Screen Class, Alert Class, Form Class, Item Class, List Class, TextBox Class, Ticker Class

Low-Level Display: Canvas: The Canvas, User Interactions, Graphics, Clipping Regions, Animation

Record Management System: Record Storage, Writing and Reading Records, Record Enumeration, Sorting Records, Searching Records, Record Listener.

JDBC Objects: The Concept of JDBC, JDBC Driver Types, JDBC Packages, Overview of the JDBC Process, Database Connection, statement Objects, Result set, Transaction Processing, Metadata, Data Types, Exceptions.

JDBC and Embedded SQL: Model Programs, Tables, Indexing, Inserting Data into Tables, Selecting Data from a Table, Metadata, Updating Tables, Deleting Data form a Table, Joining Tables, Calculating Data, Grouping and Ordering Data, Subqueries, VIEWs

Android: Introduction to Android, Architecture, memory management, communication protocols, application development methods, deployment, dealing with screen configurations and multiple screen sizes, Activity Class and its lifecycle, intents and permissions, Fragment Class, creating user interfaces, notifying users about important events, handling concurrency, acquiring data over the network, Mobile Cloud Computing with Android (MoCCA).

Recommended books:

1. Jeff McWherter, Scott Gowell “Professional Mobile Application Development”, John Wiley & Sons.
2. J2ME: The Complete Reference, James Keogh, Tata McGraw Hill.
3. Enterprise J2ME: Developing Mobile Java Applications – Michael Juntao Yuan, PearsonEducation.
4. Beginning J2ME: From Novice to Professional, Third Edition, Sing Li, Jonathan B. Knudsen, A press,2005.
5. Jochen Schiller,“Mobile Communications”, Addison-Wesley, 2ndedition, 2004.

Course Code:	**ITPC-308**
Subject Name	**MACHINE LEARNING CONCEPTS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

This course will serve as a comprehensive introduction to various topics in machine learning. At the end of the course the students should be able to design and implement machine learning solutions to classification, regression, and clustering problems; and be able to evaluate and interpret the results of the algorithms.

Course Outcomes

1. Understanding about the basic principles, techniques and applications of machine learning.
2. Broad understanding of machine learning algorithms and their use in data-driven knowledge discovery and program synthesis.
3. Knowledge of the strengths and weaknesses of different machine learning algorithms (relative to the characteristics of the application domain).
4. The ability to adapt or combine some of the key elements of existing machine learning algorithms to design new algorithms as needed.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-308												
CO 1		M	M		M			M			H	
CO 2		M	H	M	M				H	H	M	
CO 3					H	M			H		M	
CO 4		H		H		M		M		M		M

Course Content

Introduction: Well-Posed Learning Problems, Designing a Leaning System, Perspectives and Issues in Machine Learning.

Concept Learning and the General-to-Specific Ordering: Introduction, A Concept Learning Task, Concept Learning as Search, FIND-S: Finding a Maximally Specific Hypothesis, Version Spaces and the CANDIDATE-ELIMINATION Algorithm.

Decision Tree Learning: Introduction, Decision Tree Representation, Appropriate problem for Decision tree Learning, The Basic Decision Tree Learning Algorithm, Hypothesis Space Search in Decision Tree Leaning, Inductive Bias in Decision Tree Leaning, Issues in Decision Tree Leaning

Artificial Neural Networks: Introduction, Natural Network Representations, Appropriate Problems for Neural Network Learning, Perceptions, Multilayer Network and the BACKPROPAGATION Algorithm

Bayesian Learning: Introduction, Bayes Theorem, Bayes Theorem and Concept Learning, Bayes Optimal Classifier, Native Bayes Classifier, An Example: Learning to Classify Text.

Instance- Based Learning: Introduction, K-NEAREST NEIGHBOUR Learning, Distance-Weighted NEAREST NEIGHBOUR Algorithm.

Genetic Algorithms: Motivation, Genetic Algorithms, Hypothesis Space Search, Genetic Programming, Parallelizing Genetic Algorithms.

Learning Sets of Rules: Introduction, Sequential Covering Algorithms, Learning Rule Sets: Summary, Learning First-Order Rules, Learning Sets of First-Order Rules: FOIL, Induction as Inverted Deduction, Inverted Resolution.

Support Vector Machine: Maximum margin linear separators, Quadratic Programming Solution to finding maximum margin separators, Kernels for learning non-linear functions.

Recommended Books

1. Tom Mitchell, “Machine Learning”, McGraw-Hill, 1997.
2. E. Alpaydin, “Machine Learning”, MIT Press, 2010.
3. K. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
4. C. Bishop, “Pattern Recognition and Machine Learning, Springer”, 2006.

Course Code:	**ITPC-322**
Subject Name	**SOFT COMPUTING CONCEPTS LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objective

- Understand and describe soft computing techniques and their roles in building intelligent machines

List of Practicals

1. Program for implementing linear saturating function.
2. Study and analysis of art model.
3. program for error back propagation algorithm learning
4. Analysis of CPN
5. Analysis of genetic algorithm life cycle.
6. Analysis of fuzzy vs crisp logic.
7. Program and implementation of perceptron training algorithm.
8. Program to implement hebb's rule
9. Program to implement of delta rule
10. Program to implement logic gates

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code:	**ITPC-324**
Subject Name	**OBJECT-ORIENTED MODELING AND DESIGN WITH UML LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objective

- To familiar the students with alternative development processes
- To train the students for group/team projects and presentations.
- Be exposed to technical writing and oral presentations

List of Practicals

1. Understanding Problem Statements
2. Preparing Software Requirement Specification Document
3. Introduction to CASE tool: Rational Rose
4. Preliminary Use Case Diagrams
5. Detailed Use Case Diagrams
6. Class Diagrams
7. Object Diagrams
8. Activity Diagrams
9. Sequence Diagrams
10. Collaboration Diagrams
11. Deployment Diagrams
12. Practicing Analysis and Design Case Tools

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code:	**ITPC-326**
Subject Name	**MOBILE APPLICATION DEVELOPMENT LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

In this lab, a student is expected to design, implement, document and present a mobile client/server system using standard Java and Java 2 Micro Edition (J2ME) platform. Specifically it is required to design and implement a system that consists mainly of a mobile client (MC) and a Proxy Server (PS).

List of Practicals

1. Installation of Java Wireless Toolkit (J2ME)
2. Working with J2ME Features
   a. Create a program which creates a menu
   b. Event Handling
   c. Input Checking
3. Develop an application that uses GUI components, Font and Colours
4. Develop an application that uses Layout Managers and event listeners.
5. Develop a native calculator application.
6. Write an application that draws basic graphical primitives on the screen.
7. Develop an application that makes use of database.
8. Develop an application that makes use of RSS Feed.
9. Implement an application that implements Multi threading
10. Threads & High Level UI
    a. Implement an application that uses Multi threading
    b. Creating a slide show
    c. Create a MIDP application, which show to the user 5-10 quiz questions.
    d. Create a MIDP application, where the user can enter player name and points. The program saves the information to the record using RMS at MIDP device. Program should also print out the top 10 player list to the end user.
11. Develop a native application that uses GPS location information.
12. Implement an application that writes data to the SD card.
13. Implement an application that creates an alert upon receiving a message.
14. Developing Networked Applications using the Wireless Toolkit
    a. Creating a Simple Client-Server Application
    b. Creating the Datagram Server project
    c. Creating and Compiling the Datagram Server source files
    d. Running your Server application on the Phone simulator
    e. Creating the Datagram Client project
    f. Creating and Compiling the Datagram Client source files
    g. Running your Client application on the Phone simulator
15. Authentication with a Web Server
    a. Write a sample program to show how to make a SOCKET Connection from j2me phone.
    b. Login to HTTP Server from a J2ME Program
16. Web Application using J2ME
17. Android App development
    a. working with screen configurations and multiple screen sizes
    b. working with the all-important Activity Class and its lifecycle, implement intents and permissions
    c. running multiple activities with the Fragment Class, creating user interfaces

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

SEVENTH SEMESTER

S.No.	Sub. Code	Subjects	L	T	P	Credits	
1	ITPC-401	Software Testing	3	0	0	3	PC
2	ITPC-403	Cloud Computing	3	0	0	3	PC
3	ITPE-XXX	DE – III	3	0	0	3	PE
4	ITPE-XXX	DE – IV	3	0	0	3	PE
5	ITOE-XXX	OE – II	3	0	0	3	OE
Practical /Training /Projects							
6	ITPC-421	Software Testing Lab	0	0	2	1	PC
7	ITPC-423	Cloud Computing Lab	0	0	2	1	PC
8	ITCI-401	Industrial Practical Training	0	0	8	2	CIC
9	ITCI-400	Major Project (Phase I)	0	0	4	0*	CIC
TOTAL			**15**	**0**	**16**	**19**	

Course Code:	**ITPC-401**
Subject Name	**SOFTWARE TESTING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Investigate the reason for bugs and analyze the principles in software testing to prevent and remove bugs.
- Implement various test processes for quality improvement
- Design test planning.
- Apply the software testing techniques
- Use practical knowledge of a variety of ways to test software and an understanding of some of the tradeoffs between testing techniques

Course Outcomes

1. Basic software debugging methods.
2. White box testing methods and techniques.
3. Black Box testing methods and techniques.
4. Designing test plans.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-401												
CO 1			L			M		M				
CO 2		M		L					M			
CO 3				L		M	L					
CO 4		M						M				

Course Content

Introduction: Faults, errors, and failures, basics of software testing, testing objectives, principles of testing, Requirements, behavior and correctness, Testing and debugging, Test metrics and measurements, Verification, validation and testing, types of testing, software quality and Reliability, Software defect tracking.

White Box and Black Box Testing: White box testing, static testing, static analysis tools, structural testing: unit/code functional testing, code coverage testing, code complexity testing, black box testing, requirements based testing, boundary value analysis, Equivalence partitioning, state/graph based testing, Model based testing and model checking, Differences between white box and Black box testing.

Integration, system, and Acceptance Testing: Top down and bottom up integration, Bi-directional integration, system integration, Scenario Testing, Defect Bash, Functional versus Non-functional testing, Design/Architecture verification, Deployment testing, Beta testing, Scalability testing, Reliability testing, Stress testing, Acceptance testing: acceptance criteria, test cases selection and execution,

Test Selection & Minimization for Regression Testing: Regression testing, regression test process, initial smoke or sanity test, selection of regression tests, execution trace, Dynamic slicing, test minimization, tools for regression testing, ad hoc Testing: Pair testing, Exploratory testing, Iterative testing, Defect seeding.

Test Management and Automation: Test planning, management, execution and Reporting, software test automation: scope of automation, design & architecture for automation, generic requirements for test tool framework, Test tool selection, Testing in Object Oriented Systems.

Recommended books:

1. Aditya P. Mathur, “Fundamentals of Software Testing”, Pearson Education.
2. Naik and Tripathy, “Software Testing and Quality Assurance”, Wiley.
3. K. K. Aggarwal and Yogesh Singh, “Software Engineering”, New Age International Publication.
4. S. Desikan and G. Ramesh, “Software Testing: Principles and Practices”, Pearson Education.

Course Code:	**ITPC-403**
Subject Name	**CLOUD COMPUTING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- To study how to use Cloud Services.
- To implement Virtualization
- To implement Task Scheduling algorithms.
- Apply Map-Reduce concept to applications.
- To build Private Cloud.
- Broadly educate to know the impact of engineering on legal and societal issues involved

Course Outcomes

1. To understand and apply the knowledge of systems protocols and mechanisms to support Cloud computing.
2. To analyze and implement the concepts of Cloud Computing.
3. To deploy applications on cloud middleware.
4. To understand and resolve security issues in various applications.

Course Outcomes	**Program outcomes**											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITPC-403												
CO 1	L	H		M								
CO 2	L	M	M					M				
CO 3	L	M	M			H				H		
CO 4	L	M	M		M							

Course content

Cloud Computing, History of Cloud Computing, Cloud Architecture, Cloud Storage, Why Cloud Computing Matters, Advantages of Cloud Computing, Disadvantages of Cloud Computing, Companies in the Cloud Today, Cloud Services

Web-Based Application, Pros and Cons of Cloud Service Development, Types of Cloud Service Development, Software as a Service, Platform as a Service, Web Services, On-Demand Computing, Discovering Cloud Services Development Services and Tools, Amazon Ec2, Google App Engine, IBM Clouds

Centralizing Email Communications, Collaborating on Schedules, Collaborating on To-Do Lists, Collaborating Contact Lists, Cloud Computing for the Community, Collaborating on Group Projects and Events, Cloud Computing for the Corporation

Collaborating on Calendars, Schedules and Task Management, Exploring Online Scheduling Applications, Exploring Online Planning and Task Management, Collaborating on Event Management, Collaborating on Contact Management, Collaborating on Project Management, Collaborating on Word Processing - Collaborating on Databases, Storing and Sharing Files

Collaborating via Web-Based Communication Tools, Evaluating Web Mail Services, Evaluating Web Conference Tools, Collaborating via Social Networks and Groupware, Collaborating via Blogs and Wikis

Recommended books:

1. Michael Miller, Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, Que Publishing, August 2008.
2. Haley Beard, Cloud Computing Best Practices for Managing and Measuring Processes for On-demand Computing, Applications and Data Centers in the Cloud with SLAs, Emereo Pty Limited, July 2008.
3. Thomas Erl and Ricardo Puttini, "Cloud Computing: Concepts Technology & Architecture", PHI.
4. Kailash Jayaswal and Jagannath Kallakurchi, "Cloud Computing Black Book", Willey India.
5. Buyya and Vecchiola, "Mastering Cloud Computing", Morgan Kaufmann *Publishers.*

Course Code:	**ITPC-421**
SUBJECT NAME	**SOFTWARE TESTING LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

- To apply practical knowledge of a variety of ways to test software and an understanding of some of the tradeoffs between testing techniques.

List of Practicals

1. Understand The Automation Testing Approach (Theory Concept)
2. Conduct a test suite for any two web sites.
3. Test a program to login a specific web page.
4. Test a program to update 10 student records into table into Excel file.
5. Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects).
6. Program to provide total number of objects present / available on the page .
7. Implement a program to get the number of list items in a list /combo box.
8. Program to count number of check boxes on the page checked and unchecked count.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code:	**ITPC-423**
SUBJECT NAME	**CLOUD COMPUTING LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

- Understand the basic principle and architecture of cloud computing in different environments.
- To understand Virtualization Basics, Objectives of Virtualization, and Benefits of Virtualization in cloud
- An introduction about identity management in cloud and simulate it by using OpenStack
- Understand how to create, manage user and group of users accounts

LIST OF PRACTICALS

1. Installation & configuration of Oracle Virtual box for windows XP & android.
2. Installation configuration of Hadoop.
3. Using Hadoop for counting word frequency with map reduce.
4. Service deployment research & uses over cloud-Google app & Amazon web services.
5. Demonstration of Cloud Monitoring tool.
6. Performance evaluation of services over cloud-Google App & Amazon web services.
7. Service deployment and usage over cloud.
8. Managing cloud computing resources.
9. Using existing cloud characteristics and service models.
10. Performance evaluation of services over cloud.
11. Evaluate the case study: Google App Engine, Microsoft Azure
12. Evaluate the case study: Hadoop, Amazon, Aneka, MAP Reduce & HDFS.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

EIGHTH SEMESTER

S.No.	Sub. Code	Subjects	L	T	P	Credits	
1	ITPC- 402	Introduction to System Programming	3	0	0	3	PC
2	ITPC- 404	E- Commerce	3	0	0	3	PC
3	ITPC- 406	Decision Support Systems Methodology	3	0	0	3	PC
4	ITPE-XXX	DE – V	3	0	0	3	PE
5	ITOE-XXX	OE – III	3	0	0	3	OE
Practical /Training /Projects							
6	ITPC-422	Introduction to System Programming Lab	0	0	2	1	PC
7	ITCI-402	Industrial Lecture	0	0	2	1	CIC
8	ITCI-400	Major Project (Phase II)	0	0	8	4*	CIC
TOTAL			**15**	**0**	**12**	**21**	

Course Code:	**ITPC-402**
Subject Name	**INTRODUCTION TO SYSTEM PROGRAMMING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- To introduce students the concepts and principles of system programming and to enable them to understand the duties and scope of a system programmer.
- To provide students the knowledge about both theoretical and practical aspects of system programming, teaching them the methods and techniques for designing and implementing system-level programs.
- To train students in developing skills for writing system software with the aid of sophisticated OS services, programming languages and utility tools.

Course Outcomes

1. Enumerate and explain the function of the common operating system kernel routines that are provided by an operating system and accessible from a systems programming language.
2. Design, write, and test moderately complicated low-level programs using a systems programming language.
3. Create a program that measures or simulates performance and use it to analyze behavior.

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITPC-402												
CO 1	H		M	H			M		H			
CO 2			H	H			M					
CO 3			H	H		M		H				

Course content

Introduction: Introduction to Software processors, Translators and Loaders, Interpreters

Assemblers: Elements of Assembly Language Programming, Design of Two-Pass assemblers

Macros and Macro Processors: Macro Instructions, Features of a Macro facility, Implementation of Two pass Macro.

Compilers: Aspects of Compilation, Phases of compilation, Scanning and Parsing, Compilation of Expressions, Compilation of Control Structures Code Generation and Code optimization techniques, Compiler Writing Tools

Loaders & Linkage Editors: Loading Linking and Relocation, Overview of Linkage Editing, Linking for Program Overlay.

Editors and debuggers: introduction to editors, types of editor, design of an editor, debug monitors, introduction to various debugging techniques, turbo c++ debuggers.

Grammar and automation: introduction to grammar, types of grammar, acceptability of grammar, introduction to automation, characteristics of automation, finite control, transition system, finite automation. Case study on LEX and YACC.

Introduction to Operating systems: Introduction, Operating System Structures, Process Management, Memory management, I/O systems, Distributed Operating Systems

RECOMMENDED BOOKS:

1. Beck L L, “Systems Software: An Introduction to Systems Programming”, Addison-Wesley 2001.
2. Donovan J J, “Systems Programming “, New York, Mc-Graw Hill 1972.
3. Dhamdhere, D M, “Introduction to Systems Software”, Tata Mc-Graw Hill 2000.
4. Glingaert P, “Assembles Loaders and Compilers”, prentice Hall 1972.
5. Aho A V and J D Ullman, “Principles of compiler Design”, Addison Wesley/ Narosa 1985.

Course Code:	**ITPC-404**
Subject Name	**E-COMMERCE**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Acquaint students with a fundamental understanding of the environment and strategies in the New Economy.
- Provide analytical tools to understand opportunities in unserved or underserved New Economy markets.
- Provide a fundamental understanding of the different types and key components on business models in the New Economy.
- Provide guiding principles behind the design and strategy of the customer web interface.
- Understand the traditional and new communication/marketing approaches that create competitive advantage in the New Economy.
- Provide insights on how to implement strategy in the New Economy.
- Understand the metrics that New Economy firms to use to measure progress, customer satisfaction, and financial performance.

Course Outcomes

1. Understand the components and roles of the Electronic Commerce environment.
2. Describe the qualities of an effective Web business presence.
3. Describe E-Commerce payment systems.
4. Understand Web marketing approaches and elements of branding.
5. Understand legal and ethical issues related to E-Commerce

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITPC-404												
CO1	H	M		M								
CO2	M	M										
CO3	M	M	H	L								
CO4	M	M	H	L								
CO5								L		L		

Course Content

Introduction: definition of electronic commerce, e-commerce: technology and prospects, incentives for engaging in electronic commerce, needs of e-commerce, advantages and disadvantages, framework, Impact of E-commerce on business, E-Commerce Models.

Network Infrastructure for E- commerce: Internet and intranet based E-commerce- Issues, problems and prospects, Network infrastructure, network access equipments, Broadband telecommunication (ATM, iSdn, frame Relay).

Mobile commerce: introduction, Wireless application protocol, wap technology, mobile Information device.

Web security: security issues on web, importance of firewall, components of firewall, transaction security, Emerging client server, Security threats, network security, factors to consider in firewall design, Limitation of Firewalls.

Encryption: encryption techniques, symmetric encryption: keys and data encryption standard, Triple encryption, secret key encryption; asymmetric encryption: public and private pair key encryption, Digital Signatures, virtual private Network.

Electronic payments: overview, the set protocol, payment gateway, certificate, digital Tokens, smart card, credit card, magnetic strip card, e-checks, credit/debit card based EPS, online Banking.

EDI application in business, e- commerce law, forms of agreement, govt. Policies and

Agenda.

Recommended books:

1. Ravi Kalakota, Andrew Winston, “Frontiers of Electronic Commerce”, Addison- Wesley.
2. Pete Lohsin , John Vacca “Electronic Commerce”, New Age International
3. Goel, Ritendra “E-commerce”, New Age International
4. Laudon, “E-Commerce: Business, Technology, Society”, Pearson Education
5. Bajaj and Nag, “E-Commerce the cutting edge of Business”, TMH
6. Turban, “Electronic Commerce 2004: A Managerial Perspective”, Pearson Education

Course Code:	**ITPC-406**
Subject Name	**DECISION SUPPORT SYSTEMS METHODOLOGY**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- To understand how Decision Support Systems (DSS) can help management professionals;
- To analyze and to evaluate Decision Support Systems (DSS) contributions to organizations competitiveness, efficiency and quality;
- To apply proper DSS to specific business contexts;
- To be able to manage and to develop DSS business applications.

Course Outcomes

1. Understand different types of Decision Making strategies, frame work for decision support.
2. Examine DSS Development Methodology, DSS Technology Levels and Tools.
3. Analyze Group Decision Making, Communication and Collaboration, Communication Support.
4. Describe the evolution of Executive and Enterprise Information Systems, Executive’s roles and information needs, Characteristics and capabilities of Executive Support Systems.

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2
ITPC-406												
CO 1.	H	M		M								
CO 2.	M	M										
CO 3.	M	M	H	L								
CO 4.	M	M	H	L								

Course Content

Decision-making and Computerized Support Management Support Systems: An Overview, Decision-Making Systems, Modeling, and Support.

Decision Support Systems: An Overview Modeling and Analysis.

Business Intelligence: Data Warehousing, Data Acquisition, Data Mining, Business Analytics, and Visualization. Decision Support System Development.

Collaboration, Communication, enterprise decision support systems, and knowledge management, Collaborative

Computing Technologies: Group Support Systems. Enterprise Information Systems, Knowledge Management.

Intelligent decision support systems, Artificial Intelligence and Expert Systems. Knowledge-Based System, Knowledge Acquisition, Representation, and Reasoning.

Advanced Intelligent Systems, Intelligent Systems Over the Internet.

Implementing MSS in the e-Business Era, Electronic Commerce. Integration, Impacts, and the Future of Management-Support Systems.

Recommended books:

1. George M. Marakas, "Decision Support Systems in 21st century", 2nd edition phi.
2. Efraim Turban, Jay E. Aronson, Ting-Peng Liang, "Decision Support Systems and Intelligent Systems", 6th edition PHI.
3. J. L. Bennett, "Buliding Decision Support System", Addison Wesley Publications, 2002.
4. Sprague and Watson, "Decision Support Systems : Theory and Practice", PHI, 2002.
5. R. Jaya shankar, "decision support systems", tata mcgraw hill, 2002

Course Code:	**ITPC-422**
SUBJECT NAME	**INTRODUCTION TO SYSTEM PROGRAMMING LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Objectives

- To work effectively in a UNIX-style environment.
- To explain the basic operations that are performed from the time a computer is turned on until a user is able to execute programs.
- Implement routines that read and write structured binary fies such as word processing documents, index systems, or serialized hierarchical data
- Design and implement system-level applications for open-source operating systems

List of Practicals

1. Identify the basic components of an operating system, describe their purpose, and explain how they function.
2. Write, compile, debug, and execute C programs that correctly use system interfaces provided by UNIX (or a UNIX-like operating system).
3. List UNIX system calls, and invoke them correctly from within C programs.
4. Describe the difference between programs, processes, and threads.
5. Explain the meaning and purpose of process control blocks and other mechanisms that the operating system uses to implement the process and thread abstractions.

6. Write, compile, debug, and execute C programs that create, manage and terminate processes and threads on UNIX.
7. Define concurrency and explain the problems that may arise because of concurrent execution of multiple processes or threads. Explain how these problems can be avoided. Write code that avoids these problems.
8. Define semaphores, mutexes, and other synchronization primitives, explain their purpose, and describe their internal implementation.
9. Describe possible problems that arise from improper use of synchronization primitives (such as deadlocks) and present their solutions.
10. Write, compile, debug, and execute C programs that use UNIX synchronization primitives.
11. Describe operating system scheduling and use UNIX interfaces to set and modify scheduling policy parameters.
12. Define UNIX signals and signal handlers, and describe their use.
13. Write, compile, debug, and execute C programs with processes and threads that interact by invoking and catching signals.
14. Describe, configure, and use operating system timers and clocks.
15. Identify and apply principles of queueing theory to evaluate system performance.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

LIST OF DEPARTMENTAL ELECTIVES

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITPE-051	Advanced Concepts in Operating System	3	0	0	3
2	ITPE-052	Wireless Data Networks	3	0	0	3
3	ITPE-053	Information Security System	3	0	0	3
4	ITPE-054	Mobile Computing	3	0	0	3
5	ITPE-055	Software Project Management Concepts	3	0	0	3
6	ITPE-056	Agile Software Development	3	0	0	3
7	ITPE-057	Principles of Compiler Design	3	0	0	3
8	ITPE-058	Principles of Programming Languages	3	0	0	3
9	ITPE-059	Data Analytics	3	0	0	3
10	ITPE-060	Digital Image Processing	3	0	0	3
11	ITPE-061	Multicore Programming	3	0	0	3
12	ITPE-062	Cyber Forensic	3	0	0	3
13	ITPE-063	Artificial Intelligence Concepts	3	0	0	3
14	ITPE-064	Internet of Things (IoT) Concepts	3	0	0	3
15	ITPE-065	Mobile Database Management System	3	0	0	3
16	ITPE-066	Advanced Computer Networks	3	0	0	3
17	ITPE-067	Fundamental of Blockchain Technology	3	0	0	3
18	ITPE-068	High Performance Computing	3	0	0	3
19	ITPE-069	Advanced Computer Architecture	3	0	0	3
20	ITPE-070	Natural Language Processing	3	0	0	3

Course Code:	**ITPE-051**
Subject Name	**Advanced Concepts in Operating System**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Define, explain, and apply operating systems concepts: process management, CPU scheduling, synchronization, memory management, file system, and the like.
- Use the operating system interface.
- Gain experience in implementing and debugging operating system components, including the kernel module, system call, synchronization primitives, and the file system.

Course Outcomes

1. Analyze the working of an operating system and its components.
2. Define and analyze the synchronization process.
3. Identify the working methodology of multithreaded applications.
4. Compare and analyze different file systems being used in different operating systems.

Course Outcomes	Program outcomes										
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2
ITPE-051												
CO 1	H	H	M	M								
CO 2		M	L	L								
CO 3		M		H	M				M			
CO 4		M		M	H							

Course Content

Introduction

Architecture, Goals & Structures of O.S, Basic functions, Interaction of O. S. & hardware architecture, System calls, Batch, multiprogramming. Multitasking, time sharing, parallel, distributed & real -time O.S.

Process Management

Process Concept, Process states, Process control, Threads, Uni-processor Scheduling: Types of scheduling: Preemptive, Non preemptive, Scheduling algorithms: FCFS, SJF, RR, Priority, Thread Scheduling, Real Time Scheduling. System calls like ps, fork, join, exec family, wait.

Concurrency control Concurrency: Principles of Concurrency, Mutual Exclusion: S/W approaches, H/W Support, Semaphores, pipes, Message Passing, signals, Monitors, Classical Problems of Synchronization: Readers-Writers, Producer Consumer, and Dining Philosopher problem. Deadlock: Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, System calls like signal, kill

Memory Management Memory Management requirements, Memory partitioning: Fixed and Variable Partitioning, Memory Allocation: Allocation Strategies (First Fit, Best Fit, and Worst Fit), Fragmentation, Swapping, and Paging. Segmentation, Demand paging Virtual Memory: Concepts, management of VM, Page Replacement Policies (FIFO, LRU, Optimal, Other Strategies), Thrashing

I/O management & Disk scheduling: I/O Devices, Organization of I/O functions, Operating System Design issues, I/O Buffering, Disk Scheduling (FCFS, SCAN, C-SCAN, SSTF), RAID, Disk Cache.

Inter Process Communication Basic Concepts of Concurrency, Cooperating process, Advantage of Cooperating process, Bounded- Buffer - Shared-Memory Solution, Inter-process Communication (IPC), Basic Concepts of Inter-process Communication and Synchronization

Multi-Processor Based and Virtualization Concepts Virtual machines; supporting multiple operating systems simultaneously on a single hardware platform; running one operating system on top of another. Reducing the software engineering effort of developing operating systems for new hardware architectures. True or pure virtualization. Para virtualization; optimizing performance of virtualization system; hypervisor call interface.

Advanced Operating System Basics of Network Operating System, Server Operating System and Real Time Operating System

Recommended Books

1. Operating System Concepts, 9th edition Peter B. Galvin, Greg Gagne, Abraham Silberschatz, John Wiley & Sons, Inc.
2. Modern Operating Systems -By Andrew S. Tanenbaum (PHI)
3. Operating Systems 5th Edition, William Stallings, Pearson Education India

Course Code:	**ITPE-052**
Subject Name	**WIRELESS DATA NETWORKS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Introduce students with the wireless networking concepts.
- Introduction to wireless & Mobile network goals and network applications.
- Introduce students with network classifications: WLANs, WMANs and WWANs.
- Distinguish between different Encoding and Modulation techniques.
- Introduce students with Wireless Network Standards, such as IEEE 802.11, IEEE 802.15, and IEEE 802.16.
- Compare different wireless technologies.
- Study of Multiplexing techniques in wireless networks.
- Study of channelization in wireless networks.
- Study of various personal wireless technologies such as RFID, Zigbee, Bluetooth, NFC, etc.

Course Outcomes

1. Investigate the wireless network basic concept with cell structure, Modulation techniques, Coding techniques and Application
2. Understand what research problems sensor networks pose in disciplines such as signal processing, wireless communications and even control systems
3. Apply MAC layer protocols and guided on the protocols developed.
4. Apply Routing protocols for wireless network.

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITPE-052												
CO 1	H	M	L								H	L
CO 2	H	M	L			M			H		H	L
CO 3	H	M									H	L
CO 4	H		L	M							H	L

Course content

Basics of wireless data transmission: frequencies & regulations, signal propagation, propagation models, modeling the propagation loss, multiplexing, spread spectrum, antennas, cellular systems.

Cellular system concept: Cellular Hierarchy, System Management, Cellular Reuse Pattern, Ways of increasing the system capacity, Channel assignment to the cells

Media Access Techniques: SDMA, FDMA, TDMA, CDMA, Aloha, CSMA, BTMA, DBTMA, FAMA, PUMA, DAMA, PRMA, C-PRMA, MACA, MACA-BI, MACAW, CARMA, CSMA/CA, polling.

Wireless LANs: IEEE 802.11 a/b/e/f/g/i, HIPERLAN, HomeRF, OpenAir.

Wireless PANs: Bluetooth: IEEE 802.15,. UWB PAN Technology

Wireless MAN (IEEE 802.16): IEEE 802.16-2004(802.16d) for fixed WiMAX and 802.16(802.16e) for mobile WiMAX

Wireless Telecommunication Systems: Basic architecture and working of followings: WLL, GSM, Handover process, GPRS, EDGE, UMTS, CDMA2000, 3G and 4G Systems,

Software defined Radio: The Software Radio concept, Minimum radio standard, Basic elements of Software Radio architecture

Emerging wireless technologies for mobile data networking.

Recommended Books

1. Michel Daoud Yacoub, “Wireless Technology: Protocols, Standards, and Techniques”, CRC Press, 2001.
2. K. Wesołowski, “Mobile Communication Systems”, Wiley Publication, 2002.
3. J. Schiller, “Mobile Communications”, Addison-Wesley, 2004.
4. J. Geier, “Wireless LAN”, 2/e, SAMS, 2001.
5. G. Held, “Data Over Wireless Networks”, McGraw-Hill, 2001.

Course Code:	**ITPE-053**
Subject Name	**INFORMATION SECURITY SYSTEM**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Understanding of information and system security in contemporary distributed systems, based on popular commercial and public domain packages
- Understanding of the threats to information systems
- Understanding of the vulnerabilities in information systems
- Understanding of computer security countermeasures
- Understanding of network security

Course Outcomes

1. Analyze the vulnerabilities in any computing system and hence be able to design a security solution.
2. Identify the security issues in the network and resolve it.
3. Evaluate security mechanisms using rigorous approaches, including theoretical
4. Compare and Contrast different IEEE standards and electronic mail security

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITPE-053												
CO 1	H	M	L	L		M						
CO 2	M	H	L	H	M	L						
CO 3	L	M	H	H	M	L						
CO 4	H	L	M	H	M	L						

Course content

Introduction to information Security, Types of information security controls and purposes of Information Security Management, Allocation of information security responsibilities.

Telecommunications Security Objectives, Threats and Countermeasures, Identification of Security threats and development of countermeasures, Technologies and Security policies.

An introduction to LAN/WAN Security and internet Security , Security Management for the World Wide Web and Internet firewalls and Assessing inherent wireless network security deficiencies, Wireless LAN Problems, Wireless Equivalent Protocol Flaws, Short term solutions and Long term solutions.

Risk Management and Business Continuity Planning and Risk Analysis, Risk Analysis and Assessment, Available Standards (ISF, ISO and Commercial Sources), Risk verses standards etc.

Description of Various Security risk analysis tools, Techniques to secure networks from unauthorized activity, authentication procedures, encryption standards and implementations, ports and protocols that hackers manipulate, and how to engage in proactive detection and response/reporting methods.

Overview of IT Security, Hacking and Intrusion Attacks, Denial of Service Attacks (DoS), Viruses, how these get past the Firewall, how they work and the impact they can have on operations and business, Detection and Prevention Mechanisms, The self-Hack Audit and network security.

Current trends in breaches to IT Security, Current trends in IT Security detection and prevention, Examples of the types of IT security breaches most common and what can be expected in the future.

An overview of the ISO/IEC 17799:2000 “Information technology – Code of Practice for information security management” standards and how to apply these standards to IT environment, How is risk assessment related to ISO/IEC 17799 and BS 7799 Part 2?

Recommended Books

1. S. Cimato and C.Galdi, “Security in Communication Networks”, Springer, 2003.
2. H. Chan and V. Gligor, “Information Security”, Springer, 2002.
3. UPTEC Computer Consultancy Limited, “Information Technology Tools and Applications”, Elsevier, 2005.
4. V. Rajaraman, “Introduction to Information technology”, Prentice Hall of India, 2/e, 2013.
5. Thomas M. Thomas, D. Stoddard, “Network Security”, Pearson Education, 2/e, 2005.

Course Code:	**ITPE-054**
Subject Name	**MOBILE COMPUTING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course objectives

- To learn about the concepts and principles of mobile computing;
- To explore both theoretical and practical issues of mobile computing;
- To develop skills of finding solutions and building software for mobile computing applications.

Course Outcomes:

1. Able to understand the infrastructure to develop mobile communication systems (cellular theory) and the characteristics of different multiple access techniques in mobile communication.
2. Enables the students to visualize the various important steps in GSM communication
3. Enables the students to examine the important aspects of Mobile Adhoc Networks
4. Enables the students to analyze and compare the various wireless communication technologies.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPE-054												
CO 1		M	M									
CO 2		M		H				H				
CO 3			H	M	M							
CO 4	H	H			M	H	H		H		H	

Course content

INTRODUCTION

Introduction to Mobile Computing-Wireless transmission: Propagation, Modulation, Multiplexing, Spread Spectrum and Cellular Systems.

WIRELESS PROTOCOLS

Infrastructure and adhoc networks – IEEE 802.11: Protocol architecture – Physical and MAC layer; Hiperlan2: Reference model and configurations – Physical layer – Data link layer & Convergence layer; Bluetooth: Protocol stack – radio layer – Baseband layer – Link manager protocol – L2CAP layer and security.

WIRELESS NETWORKING

Satellite systems – Cellular networks – Cordless systems – Wireless Local Loop – IEEE 802.16: System reference model – Protocol architecture – MAC layer & Physical layer.

PACKET RADIO NETWORKS

Packet Radio Networks: Architecture and components of PRNETs – Routing in PRNETs – Pacing techniques– Media access and flow acknowledgement in PRNETs.

AD-HOC MOBILE NETWORKS

Types of Ad-hoc mobile communications & Host movements – Challenges facing Ad-hoc mobile networks – Problems in Ad-hoc channel access – Existing Ad-hoc MAC protocols: MACA – MACABI – PAMAs –

DBTMA.

Recommended Books

1. Jochen Schiller, "Mobile Communications", Pearson Education, Second Edition 2002.
2. C.K. Toh, "Ad Hoc Mobile Wireless Networks: Protocols and Systems", Pearson Education, 2002.
3. William Stallings, "Wireless Communications and Networks", Pearson Education 2002.

Course Code:	**ITPE-055**
Subject Name	**SOFTWARE PROJECT MANAGEMENT CONCEPTS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course objectives

- Deliver successful software projects that support organization's strategic goals
- Match organizational needs to the most effective software development model
- Plan and manage projects at each stage of the software development life cycle (SDLC)
- Create project plans that address real-world management challenges
- Develop the skills for tracking and controlling software deliverables

Course outcomes

1. Manage the scope, cost, timing, and quality of the project, at all times focused on project success as defined by project stakeholders.
2. Align the project to the organization's strategic plans and business justification throughout its lifecycle.
3. Identify project goals, constraints, deliverables, performance criteria, control needs, and resource requirements.
4. Implement project management knowledge, processes, lifecycle and the embodied concepts, tools and techniques in order to achieve project success.

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITPE-055												
CO 1	M	L			M						H	
CO 2	M	L		L		H					H	
CO 3	M	L	M				M				L	
CO 4	M	L		M				H				

Course content

Conventional Software Management: The waterfall model, conventional software Management performance. Evolution of Software Economics: Software Economics, pragmatic software cost estimation.

Improving Software Economics: Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections.
The old way and the new: The principles of conventional software Engineering, principles of modern software management, transitioning to an iterative process.

Life cycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases. Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts.

Model based software architectures: A Management perspective and technical perspective.
Work Flows of the process: Software process workflows, Iteration workflows.

Checkpoints of the process: Major mile stones, Minor Milestones, Periodic status assessments.
Iterative Process Planning: Work breakdown structures, planning guidelines, cost and schedule estimating, Iteration planning process, Pragmatic planning.

Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, evolution of Organizations.
Process Automation: Automation Building blocks, The Project Environment.

Project Control and Process instrumentation: The seven core Metrics, Management indicators, quality indicators, life cycle expectations, pragmatic Software Metrics, Metrics automation.
Tailoring the Process: Process discriminants.

Future Software Project Management: Modern Project Profiles, Next generation Software economics, modern process transitions.
Case Study: The command Center Processing and Display system- Replacement (CCPDS-R)

Recommended Books

1. Software Project Management, Walker Royce: Pearson Education, 2005.
2. Software Project Management, Bob Hughes and Mike Cotterell: Tata McGraw-Hill Edition.
3. Software Project Management, Joel Henry, Pearson Education.
4. Software Project Management in practice, Pankaj Jalote, Pearson Education.2005

Course Code:	**ITPE-056**
Subject Name	**AGILE SOFTWARE DEVELOPMENT**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- To explain how an iterative, incremental development process leads to faster delivery of more useful software
- To discuss the essence of agile development methods
- To explain the principles and practices of extreme programming
- To explain the roles of prototyping in the software process

Course Outcomes

1. Describe two or more agile software development methodologies.
2. Identify the benefits and pitfalls of transitioning to agile.
3. Compare agile software development to traditional software development models.
4. Apply agile practices such as test-driven development, standup meetings, and pair programming to their software engineering practices.

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITPE-056												
CO 1	H						M		H			
CO 2	M	M	H					H		L		
CO 3	M	H	M					M	M	M		H
CO 4	M	H			M					M	H	M

Course Contents

Overview of Software Engineering: Process, Project, Product, Method, Tool, Modern Life cycle, Traditional development approaches

Advanced Process models: V-Model, Component based development model, Agile Development Model, Unified Process Model, Extreme Programming, Feature Driven development, Lean Software Development, Service Oriented Architecture, Aspect Oriented Development

Agile Project Management: Agile Scrum Framework, Project Planning, Scheduling, Agile Estimation, Iterative Planning, Roles

Software Specification: New paradigms in software specification and design, Agile Specification, Short review of UML.

Design Engineering: Software architecture, Object-oriented Design, Software Patterns, Pattern-oriented Design, Component-oriented design. Software Frameworks, Agile Design

Agile Testing and Test Driven Development: The Agile lifecycle and its impact on testing, Test-Driven Development (TDD), xUnit framework and tools for TDD, Testing user stories - acceptance tests and scenarios, Planning and managing testing cycle, Exploratory testing, Risk based testing, Regression tests, Test Automation, Tools to support the Agile tester

BOOKS RECOMMENDED:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 7th edition. McGraw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson education.
3. Agile Software Development with Scrum By Ken Schawber, Mike Beedle, Pearson, 21 Mar 2008
4. Agile Software Development, Principles, Patterns and Practices By Robert C. Martin, Prentice Hall, 25 Oct 2002

Course Code:	**ITPE-057**
Subject Name	**Principles of Complier Design**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Explain what is meant by compiler.
- Explain how the compiler works.
- Describe various analysis of the source program.
- Describe the phases of compiler.
- Explain about Loaders, Linkers & Assemblers.
- Differentiate Lexical and Syntax Analysis.
- Describe various compiler construction tools

Course Outcomes

1. Understand the basic concepts and application of Compiler Design
2. Apply their basic knowledge Data Structure to design Symbol Table, Lexical Analyser , Intermediate
3. Code Generation, Parser (Top Down and Bottom Up Design) and will able to understand strength of
4. Grammar and Programming Language.
5. Understand various Code optimization Techniques and Error Recovery mechanisms.
6. Understand and Implement a Parser.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPE-057												
CO 1	H	M		L	M	L						
CO 2		H		H	M	H	M					
CO 3			M		M		M	H				
CO 4			H	M					H	H	H	M
CO 5	M	M			L		M		H			
CO 6	M	M			L		M				H	H

Course Content

Overview of the Translation Process, A Simple Compiler, Difference between interpreter, assembler and compiler. Overview and use of linker and loader, types of Compiler, Analysis of the Source Program, The Phases of a Compiler, Cousins of the Compiler, The Grouping of Phases, Lexical Analysis, Hard Coding and Automatic Generation Lexical Analyzers, Front-end and Back-end of compiler, pass structure

Lexical Analyzer : Introduction to Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, A Language for Specifying Lexical Analyzers, Finite Automata From a Regular Expression, Design of a Lexical Analyzer Generator, Optimization of DFA

Parsing Theory: Top Down and Bottom up Parsing Algorithms, Top-Down Parsing, Bottom-Up Parsing, Operator-Precedence Parsing, LR Parsers, Using Ambiguous Grammars, Parser Generators, Automatic Generation of Parsers. Syntax-Directed Definitions, Construction of Syntax Trees, Bottom-Up Evaluation of S-Attributed Definitions, L-Attributed Definitions, syntax directed definitions and translation schemes

Error Recovery

Error Detection & Recovery, Ad-Hoc and Systematic Methods Intermediate Code Generation, Different Intermediate Forms, Syntax Directed Translation Mechanisms And Attributed Mechanisms And Attributed, Definition

Run Time Memory Management Source Language Issues, Storage Organization, Storage-Allocation Strategies, and Access to Non local Names, Parameter Passing, Symbol Tables, and Language Facilities for Dynamic Storage Allocation, Dynamic Storage Allocation Techniques, Code Optimization

Global Data Flow Analysis, A Few Selected Optimizations like Command Sub Expression Removal, Loop Invariant Code Motion, Strength Reduction etc

Code Generation: Issues in the Design of a Code Generator, The Target Machine, Run-Time Storage Management, Basic Blocks and Flow Graphs, Next-Use Information, A Simple Code Generator, Register Allocation and Assignment, The DAG Representation of Basic

Blocks, Peephole Optimization, Generating Code from DAGs, Dynamic Programming Code-Generation Algorithm, CodeGenerator Generators.

Recommended Books

1. Compilers: Principles, Techniques and Tools By Aho, Lam, Sethi, and Ullman, Second Edition, Pearson, 2014
2. Compiler Design in C By Allen I. Holub, Prentice-Hall/Pearson.
3. Advanced Compiler Design and Implementation By Muchnick, Morgan and Kaufmann, 1998.

Course Code:	**ITPE-058**
Subject Name	**PRINCIPLES OF PROGRAMMING LANGUAGES**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- To introduce the major programming paradigms, and the principles and techniques involved in design and implementation of modern programming languages.
- To introduce notations to describe syntax and semantics of programming languages.
- To analyze and explain behavior of simple programs in imperative languages using concepts such as binding, scope, control structures, subprograms and parameter passing mechanisms.
- To introduce the concepts of ADT and object oriented programming for large scale software development.
- To introduce the concepts of concurrency control and exception handling.

Course Outcomes

1. Understand the fundamental concepts of most programming languages & the tradeoff between language design and implementation.
2. Assess programming languages critically and scientifically.
3. Understand different programming paradigms: analyze the principles of imperative, object-oriented, functional and logic programming.
4. Design a new programming language in principle.

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2
ITPE-058												
CO 1	H	M		L	M	L						
CO 2		H		H	M	H	M					
CO 3			M		M		M	H				
CO 4			H	M					H	H	H	M

Course Content

Introduction: Characteristics of programming Languages, Factors influencing the evolution of programming language, developments in programming methodologies, desirable features and design issues. Programming language processors: Structure and operations of translators, software simulated computer, syntax, semantics, structure, virtual computers, binding and binding time.

Elementary and Structured Data Types: Data object variables, constants, data types, elementary data types, declaration, assignment and initialization, enumeration, characters, strings. Structured data type and objects: Specification of data structured types, vectors and arrays, records, variable size data structure, pointers and programmer constructed data structure, Sets files. Sub Program and programmer defined data types: Evolution of data types, abstractions, encapsulations, information hiding, sub programmes, abstract data types.

Sequence Control; Implicit and Explicit sequence control, sequence control with within expression and statements, recursive sub programmes, exception handling, co routines, Scheduled sub programmes, concurrent execution. Data control referencing environments, static and dynamic scope, local data local data referencing environment, shared data: Explicit common environment dynamic scope parameter passing mechanism.

Storage Management: Major run time requirements, storage management phases, static storage management, stack based, heap based storage management. Syntax and translation: General syntactic criteria, syntactic element of a language, stages in translation, formal syntax and semantics.

Operating and Programming Environment: Batch Processing Environments, Embedded system requirements, Theoretical models, Introduction to Functional Programming, Lambda calculus, Data flow language and Object Oriented language, Comparison in various general and special purpose programming languages e.g. Fortran, C, Pascal, Lisp, etc.

Recommended Books

1. Terrance W Pratt, "Programming Languages: Design and Implementation" PHI
2. Sebesta, "Concept of Programming Language", Addison Wesley
3. E Horowitz, "Programming Languages", 2nd Edition, Addison Wesley
4. "Fundamentals of Programming Languages", Galgotia.
5. Louden, “programming Languages-principles and practice”, Cengage Learning, New Delhi

Course Code:	**ITPE-059**
Subject Name	**DATA ANALYTICS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course objectives

- Discuss the overall process of how data analytics is applied
- Discuss how data analytics can be used to better address and identify risks
- Demonstrate the power of data analytics using case studies

Course outcomes

1. Understand what Big Data is and why classical data analysis techniques are no longer adequate.
2. Understand the benefits that Big Data can offer to businesses and organizations.
3. Learn conceptually how Big Data is stored.
4. See how Big Data can be analyzed to extract knowledge.

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2
ITPE-059												
CO 1	M	H	L				L				H	H
CO 2	H	L	M			M			H		M	H
CO 3	M	L	M				L				M	H
CO 4	H	M	H		L				L		M	H

Course content

Introduction: Big Data Overview, Challenges with Big Data, Traditional Business Intelligence (BI) versusBig Data.

Big Data Analytics: The rising and importance of data sciences, Classification of Analytics, Data

Sciences,Terminologies in Big Data Environment (In –Memory Analytics,In-Database Processing, SymmetricMultiprocessor system, Massively parallel processing, Parallel and distributed system, Shared nothingarchitecture, CAP Theorem), Analytics tools.

Hadoop Overview: Introduction to Hadoop, RDBMS vsHadoop, key aspects of hadoop, hadoopcomponents,hadoop conceptual layer, high level architecture of hadoop.

Hadoop Architecture: Hadoop architecture, Hadoop ecosystem components, Hadoop storage: HDFS,Hadoop processing, Map Reduce Framework, Hadoop server roles.

Hadoop big data technology landscape: NoSQL, Types of NoSQL database, Advantages, New SQL,Comparison of SQL, NoSQL and NewSQL.

Overview of Data Analytics- Theory and Methods: Measures and evaluation, Supervised Learning, Linearand Logistic Regression, Decision trees, Unstructured data analytics.

Recommended Books

1. Big data and Analytics by SeemaAcharya and SubhashiniChellappan.
2. Hadoop: The Definitive Guide by Tom White.
3. Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQLLandGraph by David Loshin.

Course Code:	**ITPE-060**
Subject Name	**DIGITAL IMAGE PROCESSING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course objectives

- To study the image fundamentals and mathematical transforms necessary for image processing.
- To study the image enhancement techniques
- To study image restoration procedures.
- To study the image compression procedures.

Course outcomes

1. Review the fundamental concepts of a digital image processing system.
2. Analyze images in the frequency domain using various transforms.
3. Evaluate the techniques for image enhancement and image restoration.
4. Categorize various compression techniques.
5. Interpret Image compression standards.
6. Interpret image segmentation and representation techniques.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPE-060												
CO 1	H						M		H			
CO 2	M	M	H					H		L		
CO 3	M	H	M					M	M	M		H
CO 4	M	H			M					M	H	M
CO 5			M		M	M	M			L		M
CO 6				M	M	H	M					

Course content

Digital Image Fundamentals: Why is Computer Vision Difficult?, Different stages of image processing and analysis, Components of image processing system, Sampling and Quantization, Some basic relationships like neighbor's connectivity, distance measure between pixels.

Image Enhancement and Restoration: Basic Intensity Transformation Functions, Histogram processing, Spatial Domain methods: Fundamentals of spatial filtering, Smoothing Spatial Filters, Sharpening Spatial Filters, Frequency domain methods: low pass filtering, High pass filtering, Image Degradation/Restoration model

Image Compression: Fundamentals of image compression, error criterion, Coding Inter-pixel and Psycho visual redundancy, Image Compression models, Error free compression: Huffman, Arithmetic, Run length Coding, Lossy Compression: Block Transform Coding based on DCT and DWT, Image Compression standard: JPEG.

Morphological image processing: Basic Morphology concepts, Binary dilation and erosion, Opening and Closing operations, Basic Morphological Algorithms: Boundary extraction, Hole Filling, Extraction of Connected Components.

Image Segmentation and Edge Detection: Fundamentals, Point, Line and Edge Detection: Detection of isolated points, lines, Basic Edge Detection, Advanced Edge detection using Canny edge detector, Laplacian edge detector and Laplacian of Gaussian edge detector. Edge Linking and Boundary Detection, Thresholding: Basic Global Thresholding and Optimum Global Thresholding using Otsu's Method, Region Based Segmentation: Region Growing, Region Splitting and Merging

Representation and Description: Representation schemes like chain coding, Polygonal approximation using minimum perimeter polygon, Signatures, Boundary Descriptors: Shape Numbers, Fourier, and Statistical moments. Regional Descriptors: Topological Descriptors, Texture, Moment Invariants

Recognition and Interpretation: Pattern and pattern classes, Decision Theoretic methods: minimum distance classifier, matching by correlation, Structural Methods: Matching Shape Numbers

Recommended Books

1. Rafael C. Gonzales and Richard E. Woods, "Digital Image Processing", Pearson Education, 3/e, 2007.
2. Milan Sonka, Vaclav Hlavac and Roger Boyle, "Digital Image Processing and Computer Vision", Cengage Learning, 2007.
3. Anil K. Jain, "Fundamentals of Digital Image Processing", Pearson Education, 1988.
4. B. Chanda, "Digital Image Processing and Analysis", PHI Learning Pvt. Ltd., 2011.
5. William K. Pratt, "Digital Image Processing", Wiley-Interscience, 4/e, 2007.

Course Code:	**ITPE-061**
Subject Name	**MULTICORE PROGRAMMING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Define terminology commonly used in parallel computing, such as efficiency and speedup.
- Describe different multicore architectures and programming models.
- Implement efficient algorithms for common application kernels, such as matrix multiplication, on conventional multicore processors, GPUs, and the Cell processor.
- Given a problem, develop an efficient parallel algorithm to solve it.

- Given a parallel algorithm, analyze its time complexity as a function of the problem size and number of processors.
- Given a parallel algorithm, implement it using MPI, OpenMP, pthreads, CUDA, or a combination of these.
- Given a parallel code, analyze its performance, determine computational bottlenecks both theoretically and empirically, and optimize the performance of the code.
- Given a parallel code, debug it and fix the errors.
- Given a problem, implement an efficient and correct code to solve it, analyze its performance, and give convincing written and oral presentations explaining your achievements

Course outcomes

1. Program Parallel Processors.
2. Develop programs using OpenMP and MPI.
3. Compare and contrast programming for serial processors and programming for parallel processors.

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITPE-061												
CO 1	H	L	M							L		
CO 2	M	L			M					L		
CO 3	M	L	M		H					L		

Course Content

MULTI-CORE PROCESSORS: Single core to Multi-core architectures – SIMD and MIMD systems – Interconnection networks - Symmetric and Distributed Shared Memory Architectures – Cache coherence - Performance Issues – Parallel program design.

PARALLEL PROGRAM CHALLENGES: Performance – Scalability – Synchronization and data sharing – Data races – Synchronization primitives (mutexes, locks, semaphores, barriers) – deadlocks and livelocks – communication between threads (condition variables, signals, message queues and pipes).

SHARED MEMORY PROGRAMMING WITH OpenMP: OpenMP Execution Model – Memory Model – OpenMP Directives – Work-sharing Constructs – Library functions – Handling Data and Functional Parallelism – Handling Loops – Performance Considerations.

DISTRIBUTED MEMORY PROGRAMMING WITH MPI: MPI program execution – MPI constructs – libraries – MPI send and receive – Point-to-point and Collective communication – MPI derived datatypes – Performance evaluation

PARALLEL PROGRAM DEVELOPMENT: Case studies - n-Body solvers – Tree Search – OpenMP and MPI implementations and comparison.

Recommended Books

1. Peter S. Pacheco, "An Introduction to Parallel Programming", Morgan-Kauffman/Elsevier, 2011.
2. Darryl Gove, "Multicore Application Programming for Windows, Linux, and Oracle Solaris", Pearson, 2011

Course Code:	**ITPE-062**
Subject Name	**CYBER FORENSIC**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- To provide an understanding Computer forensics fundamentals
- To analyze various computer forensics technologies
- To provide computer forensics systems
- To identify methods for data recovery.
- To apply the methods for preservation of digital evidence.

Course Outcomes

1. Consider the security issues related to network layer and transport layer.
2. Explain security principles in the application layer.
3. Explain computer forensics.
4. Utilize forensics tools.
5. Analyze and validate forensics data.

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITPE-062												
CO 1	H	L	M									H
CO 2	M	L			M							H
CO 3	M	L	M		H							H
CO 4	M	L	H	M	M	M						M
CO 5	M	L					M	H			M	M

Course Content

Network Layer Security and Transport Layer Security

IPSec Protocol – IP Authentication Header – IP ESP – Key Management Protocol for IPsec. Transport layer Security: SSL protocol, Cryptographic Computations – TLS Protocol.

E-mail Security & Firewalls

PGP – S/MIME – Internet Firewalls for Trusted System: Roles of Firewalls – Firewall related terminology- Types of Firewalls – Firewall designs – SET for E-Commerce Transactions.

Introduction to Computer Forensics

Introduction to Traditional Computer Crime, Traditional problems associated with Computer Crime. Introduction to Identity Theft & Identity Fraud. Types of CF techniques – Incident and incident response methodology – Forensic duplication and investigation. Preparation for IR: Creating response tool kit and IR team. – Forensics Technology and Systems – Understanding Computer Investigation – Data Acquisition.

Evidence Collection and Forensics Tools

Processing Crime and Incident Scenes – Working with Windows and DOS Systems. Current Computer Forensics Tools: Software/ Hardware Tools.

Computer Forensic Analysis and Validation

Validating Forensics Data – Data Hiding Techniques – Performing Remote Acquisition – Network Forensics – Email Investigations – Cell Phone and Mobile Devices Forensics.

Recommended Books

1. John R.Vacca, “Computer Forensics”, Cengage Learning, 2005
2. Richard E.Smith, “Internet Cryptography”, 3rd Edition Pearson Education, 2008.
3. Marjie T.Britz, “Computer Forensics and Cyber Crime”: An Introduction”, 3rd Edition, Prentice Hall, 2013.

Course Code:	**ITPE-063**
Subject Name	**ARTIFICIAL INTELLIGENCE CONCEPTS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course objectives

- To provide a strong foundation of fundamental concepts in Artificial Intelligence
- To provide a basic exposition to the goals and methods of Artificial Intelligence
- To enable the student to apply these techniques in applications which involve perception, reasoning and learning

Course Outcomes

1. Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.
2. Formalize a given problem in the language/framework of different AI methods
3. Describe basic AI algorithms (e.g., standard search algorithms or resolution).
4. Design and carry out an empirical evaluation of different algorithms on a problem formalization.

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITPE-063												
CO 1	H	M				L		L				
CO 2			M	H	H							
CO 3		M		M	L				M			
CO 4	H				H							H

Course Content

Introduction: Introduction to AI: Definitions, Historical foundations, Basic Elements of AI, Characteristics of intelligent algorithm, AI application Areas

Problem solving: State space search; Production systems, search space control: depth-first, breadth-first search, heuristic search - Hill climbing, best-first search, branch and bound. Problem Reduction, Constraint Satisfaction End, Means-End Analysis

Handling uncertainty: Non-Monotonic Reasoning, Probabilistic reasoning, use of certainty factors, Fuzzy logic

Knowledge Based Systems: Proportional Logic, FOPL, Clausal Form, Resolution & Unification. Knowledge representation, acquisition, organisation& Manipulation, Semantic nets, Frames, Conceptual Dependency, Scripts & CYC.

Machine Learning: Concept of learning, Concept creation, learning automation, supervised and Unsupervised Learning, learning tasks & learning strategies, single layer & multiplayer Perceptions, Back propagation, learning by inductions, Competitive Learning, HebbianCoincidence Learning, Attractor Networks Samuel's checkers algorithm. Hopfield nets, Adaptive resonance theory

Expert Systems: Need and justification for expert systems, Basic Components & architecture of Expert systems, ES-Shells, Representing & Using Domain Knowledge, Knowledge acquisition in expert Systems. Case studies: MYCIN, RI.

Recommended Books

1. Rich and K. Knight,"Artificial Intelligence", Tata McGraw Hill.
2. George F. Luger, “Artificial Intelligence – Structures and Strategies for Complex Problem Solving", Pearson Education.
3. Russell &Norvig, "Artificial Intelligence ‘a Modern Approach", Pearson Education.
4. Dan W. Patterson, “Introduction to Artificial Intelligence and Expert Systems”, PHI.
5. E. Charnaik and D. McDermott," Introduction to Artificial Intelligence", Addison-Wesley Publishing Company.
6. Nils J. Nilson, “Principles of Artificial Intelligence”, Narosa Publishing Co.

Course Code:	**ITPE-064**
Subject Name	**INTERNET OF THINGS (IOT) CONCEPTS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

Students will be explored to the interconnection and integration of the physical world and the cyber space. They are also able to design & develop IOT Devices.

Course Outcomes

1. Able to understand the application areas of IOT
2. Able to realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks
3. Able to understand building blocks of Internet of Things and characteristics.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPE-064												
CO 1	H	M		M		L		L				
CO 2		M	L	M								
CO 3		M		H	L				M			

Course Content

Introduction & Concepts: Introduction to Internet of Things, Physical Design of IOT, Logical Design of IOT, IOT Enabling Technologies, IOT Levels.

Domain Specific IOTs: Home Automation, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health & Life Style.

M2M & System Management with NETCONF-YANG: M2M, Difference between IOT and M2M, SDN and NFV for IOT, Software defined Networking, Network Function Virtualization, Need for IOT Systems Management, Simple Network Management Protocol, Limitations of SNMP, Network Operator Requirements, NETCONF, YANG, IOT Systems management with NETCONF-YANG.

Developing Internet of Things & Logical Design using Python: Introduction, IOT DesignMethodology, Installing Python, Python Data Types & Data Structures, Control Flow, Functions, Modules, Packages, File Handling, Date/ Time Operations, Classes, PythonPackages

IOT Physical Devices & Endpoints: What is an IOT Device, Exemplary Device, Board,Linux on Raspberry Pi, Interfaces, and Programming & IOT Devices.

Recommended Books

1. Vijay Madisetti, Arshdeep Bahga, "Internet of Things A Hands-On-Approach",2014,ISBN: 978 0996025515
2. Adrian McEwen, "Designing the Internet of Things", Wiley Publishers, 2013, ISBN:978-1-118-43062-0
3. Daniel Kellmereit, "The Silent Intelligence: The Internet of Things". 2013, ISBN0989973700

Course Code:	**ITPC-065**
Subject Name	**Mobile Database Management Systems**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objectives

- This course introduces the fundamentals of mobile databases and the respective transactional models.
- The students will be able to implement and analyze database recovery methods.

Course Outcomes

At the completion of the course, students will be able to

1. Study the fundamentals of mobile databases.
2. Understand mobile transactions.
3. Apply databases with mobile transactional models.
4. Evaluate database recovery methods in mobile database systems.

Course Content

Introduction

Introduction to Mobile Databases, Components of Mobile Databases, Fully Connected Information Space, Types of Mobility.

Fundamentals of Database Processing

Conventional Database Architecture: Centralized DBMS and Distributed DBMS, Federated Database Systems, Multidatabase Systems. Data distribution methods, Database Processing: Transactions. Serialization of Transactions. Advanced Transaction Models: Workflows, Properties of Workflow Systems. Nested and Multilevel Transactions, Saga Model, Cooperative Transactions, Contract Model, Flex Transactions

Concurrency Control Mechanism:

Introduction, ways of locking data items. the Phantom Problem, Multi granularity locking, Heuristic approach in locking scheme, Non locking based schemes.

Data Processing and Mobility

Introduction, Effect of mobility on the management of data, Data Categorization, Location dependent data distribution. Effect of mobility on ACID. Effect of connectivity on transaction processing.

Transaction Management in Mobile Database Systems

Transaction execution in MDS, Mobile Transaction Model, Execution model with Reporting Transaction, Two-level Consistency Model, Pro-motion, Pre-write transaction execution model, Kangaroo Transaction model, Moflex Transaction Model, Data Consistency in Intermittent Connectivity, Mobilaction.

Mobile Database Recovery:

Introduction, Log Management in MDS, Mobile Database Recovery Scheme.

Books and References

1. Kumar Vijay, " Mobile Database Systems". John Willy & Sons, 2006.
2. Ming-Syan Chen, Panos K. Chrysanthis, Morris Sloman, 2003
3. Amjad Umar, "Mobile Computing and Wireless Communications Applications, Networks, Platforms, Architectures, and Security", NGE Solutions, Incorporated, 2004.

Course Code:	**ITPC-066**
Subject Name	**TCP/IP**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objectives

- This course introduces the fundamentals of computer networks and advanced topics that are basic blocks of networking.
- The students will get exposure to advanced networking concepts for next generation network architecture and design.

Course Outcomes

At the completion of the course, students will be able to

1. Understand packet switching networks and routing in packet switching networks with different routing algorithms.
2. Describe traffic management at packet level, flow level and flow aggregate levels of packetswitching networks.
3. Evaluate the architecture of TCP/IP and protocols associated with TCP/IP and its variants and toanalyze the network applications, network management.
4. Implement and build network and network related concepts.

Course Content

TCP/IP Protocol Stack; Network Layer Primitives – IP Addressing, IP Routing – Intra Domain Routing Protocols, Inter Domain Routing Protocols (BGP), IP Services – SNMP, ARP; Transport Layer Primitives
– Connection Establishment and Closure, Flow Control and Congestion Control at the Transport Layer, Transmission Control Protocol – Basic Features, TCP Congestion Control, Application Layer Services (HTTP, FTP, Email, DNS).

Transmission Control Protocol (TCP): TCP variants- TCP Tahoe, TCP Reno, TCP New Reno, TCP Vegas, TCP SACK, etc. and, TCP Options: NOP, MSS, Window Scale Factor, Timestamp, SACK-Permitted and SACK Options, Stream Control Transmission Protocol (SCTP): Introduction, Services, Features, Packet Format, Association, State Transition Diagram, Flow Control, Error Control, Congestion.

Internet Protocol Version 6: IPV6 Addressing: Introduction, Address Space Allocation, Global Unicast Addresses, Auto configuration, Renumbering; IPV6 Protocol: Packet Format, Transition from Ipv4 to Ipv6; Generic Routing Encapsulation (GRE). ICMPv6: Error Messages, Informational Messages, Neighbours-Discovery Messages, Group Membership Messages.

Next Generation Network: Unicast Routing Protocols: RIP, OSPF; Multicasting and Multicast Routing Protocols: Introduction, Multicast Addresses, IGMP, Multicast Routing, Routing Protocols, MBone.

Text/References:

1. Kevin R. Fall, W. Richard Stevens, “TCP/IP Illustrated”, Volume 1: The Protocols, 2nd Edition, Addison-Wesley Professional, 2011.

2. Kurose James F. and Ross Keith W., “Computer Networking: A Top-Down Approach featuring the Internet”, 6/e, 2017.

3.Larry L. Peterson & Bruce S. Davie, “Computer Network: A System Approach”, Morgan Kaufmann, 75/e, 2012.

3. Jochen Schiller, “Mobile Communications”, Pearson Addison-Wesley, 4/e, 2017.

4. Behrouz A. Forouzan, “TCP/IP Protocol Suite”, McGraw- Hill, 4/e, 2017.

5. James F. Kurose, Keith W. Ross, “Computer Networking”, Pearson, 2012. 5. Charles M. Kozierok,“The TCP/IP Guide”, No starch press, 2005.

Course Code:	**ITPC-067**
Subject Name	**Fundamentals of Blockchain Technology**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objective

This course is intended to make students:

- Study the basics of Blockchain technology.
- Understand how blockchain systems (mainly Bitcoin and Ethereum) work.
- Analyze applications and legislation for blockchain.
- Build and deploy smart contracts and distributed applications.
- Implementing learner will have idea about private and public Blockchain, and smart contract.

Course Outcomes

After the completion of this course, student will be able to

1. Understand and explore the working of Blockchain technology.
2. Analyze the working of Smart Contracts
3. Apply the learning of solidity and de-centralized apps on Ethereum.
4. Evaluate security, privacy, and efficiency of a given blockchain system.

Course Content

Introduction: Overview, Digital Age, Internet of Information, Concept of Trust, Trust protocol, blockchain technology and working principles of Blockchain transaction, Blockchain Challenges, Transactions and Blocks, P2P Systems.

Components of Blockchain: Importance of distributed consensus, Hash Puzzles, Digital Signatures, Hashing, public key cryptosystems, private vs public blockchain and use cases. Blockchain advantage over conventional distributed database, Blockchain Network, Mining Mechanism, Distributed Consensus, Merkle Tree, Transactions and Fee, Reward, Chain Policy, Life of Blockchain application, Soft & Hard Fork, Sybil Attack.

Blockchain design principles: Network integrity, Distributed Power, Value as Incentives, Security, Privacy, Rights Preservation, Inclusion, and Guidelines for choosing Blockchain project. Example case studies, Application areas.

Blockchain Implementation Challenges: Technology challenges, The Energy Consumption, Governments role, Governing the Protocols, Distributed Autonomous Agents, Privacy, Malicious usage.

Blockchain Transactions and consensus: The real need for mining – consensus – Byzantine Generals Problem, and Consensus as a distributed coordination problem, Consensus algorithms, RAFT, Paxos, Byzantine fault Tolerance, PBFT, Consensus in Bitcoin, Bitcoin block structure, block creation and storage, and Bitcoin wallets, Proof of Work, Proof of Stake, Proof of Burn.

Introduction to Bitcoin Blockchain: Introduction to digital currency, Crypto currency, Explanation of Bitcoin with concepts of blockchain, Cryptographic methods in Bitcoin, Hashing in Bitcoin, Overview of Hash puzzle in Bitcoin, Consensus in Bitcoin, Bitcoin block structure, block creation and storage, and Bitcoin wallets.

Introduction to Smart Contracts: advantage of smart contracts, Introduction to REMIX IDE, Introduction to Solidity smart contracts, Solidity structure and language syntax, Deploying and interacting with smart contracts via Remix IDE.

Text/Reference Books:

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press (July 19, 2016).
2. Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies
3. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System
4. Dr. Gavin Wood, "ETHEREUM: A Secure Decentralized Transaction Ledger,"Yellow paper.2014.
5. Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli, A survey of attacks on Ethereum smart contracts

References:

- Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press (July 19, 2016).
- Antonopoulos, Mastering Bitcoin.
- Antonopoulos and G. Wood, Mastering Ethereum.
- D. Drescher, Blockchain Basics. Apress, 2017.

Course Code:	**ITPC-068**
Subject Name	**High Performance Computing**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objective

- At the end of the course, students will be able to understand the concepts of: (1) Parallel Computing (2) New Processor Architectures, (3) Power-Aware Computing and Communication, (4) Advanced Topics on Computing.

Course Outcome

1. To study parallel models of computation such as dataflow, and demand-driven computation
2. To understand the need to achieve higher performance in modern computing systems.
3. To apply various high performance computing approaches for a given problem.
4. To evaluate the advanced topics on computing.

Course Content

Parallel Processing Concepts; Levels and model of parallelism: instruction, transaction, task, thread, memory, function, data flow models, demand-driven computation.

Parallel architectures: superscalar architectures, multi-core, multi-threaded, server and cloud; Fundamental design issues in HPC: Load balancing, scheduling, synchronization and resource management.

Operating systems for scalable HPC; Parallel languages and programming environments; Performance analysis of parallel algorithms; Fundamental limitations in HPC: bandwidth, latency and latency hiding techniques; Scalable storage systems: RAID, SSD cache, SAS, SAN; HPC based on cluster, cloud, and grid computing: economic model, infrastructure, platform, computation as service;

Accelerated HPC: architecture, programming and typical accelerated system; Power-aware HPC Design: computing and communication, processing, memory design, interconnect design, power management; Advanced topics: peta scale computing; big data processing, optics in HPC, quantum computers.

References:

1. George S. Almasi and Alan Gottlieb, "Highly Parallel Computing", Addison Wesley Longman, 2/e, 1993.
2. Kai Hwang, "Advanced Computer Architecture: Parallelism, Scalability, Programmability", McGraw Hill1993
3. David Culler Jaswinder Pal Singh, Morgan Kaufmann, "Parallel Computer Architecture: A hardware/Software Approach", 1999.
4. Kai Hwang, "Scalable Parallel Computing", McGraw Hill 1998.
5. "Principles and Practices on Interconnection Networks", by William James Dally and BrianTowles, Morgan Kauffman 2004.
6. GPU Gems 3 --- by Hubert Nguyen (Chapter 29 to Chapter 41)
7. Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, Introduction to Parallel Computing, 2nd edition, Addison-Welsey, 2003.
8. Petascale Computing: Algorithms and Applications, David A. Bader (Ed.), Chapman & Hall/CRCComputational Science Series, 2007.
9. Georg Hager, Gerhard Wellein, Introduction to High Performance Computing for Scientists and Engineers,Chapman & Hall / CRC Computational Science series, 2011.

Course Code:	**ITPC-069**
Subject Name	**Advanced Computer Architecture**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objective

- At the end of the course, students will be able to acquire knowledge about various computer architecture fundamentals at advanced level.

Course Outcomes

1. To study and analyze the concepts of parallel computing and hardware technologies.
2. To understand an elaborated idea about the different architectures, memory systems and buses.
3. To explore the importance of multiprocessor and multicomputer.

Course Content

Quantitative Principles of Computer Design: The Task of a Computer Designer, Technology and Computer Usage Trends, Cost and Trends in Cost, Measuring and Reporting Performance, Benchmarksand metrics.

Instruction Set Principles and Examples: Classification of Instruction Set Architectures, Instruction Formats and Semantics, Memory Addressing Modes, Operations in the Instruction Set, Encoding and Instruction Set, The Role of Compilers.

Advanced Pipelining and Instruction-Level Parallelism: Basic Pipeline Operations, Data and Control Pipeline Hazards, Instruction-Level Parallelism, Dynamic Scheduling- Tomasulo’s algorithm and speculative execution and Branch Prediction.

Memory-Hierarchy Design: Cache-Design, Issues, modeling and optimization, Performance Evaluation, Virtual Memory Addressing, Memory Protection Mechanisms, Memory coherency techniques.

Storage Systems: Types of Storage Devices, Buses-Connecting I/O Devices to CPU/Memory, I/O Performance Measures, Reliability, Availability, and RAID, Interfacing to an Operating System.

Interconnection Networks: Interconnection network Media, Connecting More Than Two Computers, Practical Issues for Commercial Interconnection Networks, Examples of Interconnection Networks.

Multiprocessors (Time Permitting): Characteristics of Application Domains, Centralized Shared-Memory Architectures, Distributed Shared-Memory Architectures, Tiled chip multi-core processor (TCMP), Routing techniques in network on chip (NoC), NoC micro architecture, Execution Synchronization, Models of Memory Consistency.

Introduction to embedded systems and microcontrollers, sensors and actuators.

Text/References:

1. Kai Hwang and Naresh Jotwani, "Advanced Computer Architecture: Parallelism, Scalability, Programmability", 3e, 2016.
2. John L. Hennessy and David A. Patterson, "Computer Architecture: A Quantitative Approach", Morgan Kaufmann, 5/e, 2011.
3. William Stallings, "Computer Organization and Architecture", Prentice Hall, 9/e, 2012
4. Sajjan G. Shiva, "Advanced Computer Architectures", Taylor & Francis, 2018.

Course Code:	**ITPC-070**
Subject Name	**Natural Language Processing**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Outcome

- At the end of the course students will be able to learn how one could develop natural languageunderstanding models from regularities in large corpora of natural language texts.

Course Objective

1. To understand the mathematical foundations, Probability theory with Linguistic essentials such as syntactic and semantic analysis of text.
2. To analyze various natural language concepts and language modelling technique based on the structure of the language.
3. To familiarity to linguistics and their application to part-of-speech tagging.
4. To analyze and develop a real world applications based on natural language cocepts.

Course Content

Introduction: NLP introduction, origins of NLP, Language and Knowledge, The challenges of NLP, Different levels of NLP; Text Normalization: Basic pre-processing, Word and sentence segmentation, Lemmatization, Stemming, Morphology;

Language Models: n-gram models, smoothing techniques; Sequence Learning Tasks and Models, Language and Grammar, Processing Indian Languages, NLP applications, Some successful Early NLP systems, Information Retrieval Language Modeling: Introduction, Various Grammars- based language models, Statistical Language Model. Word Level Analysis: Introduction, Regular Expressions, Finite State Automata, Morphological Parsing, Spelling Error Detection and Correction, Words and Word Classes, Part-of-Speech Tagging.

Syntactic Parsing: Regular and Context-Free Languages, Context-Free Parsing, CKY Algorithm; Dependency Parsing: Dependency Grammar, Graph-based dependency parsing, Transition-based dependency parsing;

Vector-space Models: Word and Meanings, Distributional and Distributed Semantics, Lexical Semantic Analysis, GloVe, word2vec; Reference Resolution; Applications,.

Semantic Analysis: Introduction, Meaning Representation, Lexical Semantics, Ambiguity, Word Sense Disambiguation. Discourse Processing: Introduction, Cohesion, Reference Resolution, Discourse Coherence and Structure. Natural Language Generation: Introduction, Architecture of NLG Systems, Generation Tasks and Representations, Application of NLG.

Text/References:

1. Tanveer Siddiqui and U. S. Tiwary, "Natural Language Processing and Information Retrieval", Oxford Higher Education, 2008.
2. James Allen, "Natural Language Understanding", 2/e, Pearson Education, 1994.
3. D. Jurafsky and J. H. Martin, "Speech and Language Processing", Prentice Hall, 2/e, 2008.
4. L.M. Ivansca and S. C. Shapiro, "Natural Language Processing and Language Representation", AAAI Press, 2000.
5. Akshar Bharti, Vineet Chaitanya and Rajeev Sangal, "NLP: A Paninian Perspective", Prentice Hall, New Delhi, 2004
6. J. Eisenstein, Introduction to Natural Language Processing, MIT Press, 2019.

LIST OF OPEN ELECTIVES

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITOE-001	Fundamentals of Software Engineering	3	0	0	3
2	ITOE-002	Web Design Concepts	3	0	0	3
3	ITOE-003	Fundamentals of Data Analytics	3	0	0	3
4	ITOE-004	Agile Software Engineering	3	0	0	3
5	ITOE-005	Mobile Application Development Concepts	3	0	0	3
6	ITOE-006	Fundamentals of Cloud Computing	3	0	0	3
7	ITOE-007	Machine Learning	3	0	0	3
8	ITOE-008	Fundamentals of Artificial Intelligence	3	0	0	3

Course Code:	**ITOE-001**
Subject Name	**FUNDAMENTALS OF SOFTWARE ENGINEERING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Study the current software engineering techniques and examines the software life-cycle, including software specification, design implementation, testing and maintenance.
- Present software engineering methodologies for the development of Quality, cost-effective, schedule-meeting software.
- Develop an understanding of ethical and professional issues related to Software Project Delivery.

Course Outcomes

1. Able to apply the concepts and choose an appropriate SDLC process model for user requirements.
2. To analyze requirement techniques like Data flow diagram, Entity relationship diagram etc.
3. Understanding the concept of Software Design and emphasizing upon various software metrics used for analyzing the software.
4. Demonstrate various testing methodologies and debugging tools for a prototype software.
5. Design various software reliability measures to assess the quality of software in case of various faults and failures.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITOE-001												
CO 1.	M	M				L						
CO 2.		H			L					L		
CO 3.	H		H		M							
CO 4.			L	H	M				M			
CO 5.		M	M		L							

Course content

Introduction to Software Engineering: What is Software Engineering?, Why Software Engineering, Software Crisis, Notable Changes in Software Development Practices, Software myths.

The Software Process: Plan-driven and agile processes, different development philosophies: sequential vs iterative, software development life cycle (SDLC), overview of various SDLC models/methodologies. Plan-Driven Development: The Waterfall Model, Incremental development, Integration and configuration, How to Cope with changes?, Software prototyping vs Incremental delivery.

Agile-Software Development: Agility, Extreme Programming, Agile Project Management (Scrum), Scaling out and scaling up, Problem with Agile methods, Combining Agile and plan-driven methods, Agile methods across organizations Process models: The waterfall model, Incremental process models, Evolutionary process models, The Unified process.

Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

Requirements Engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

Software Design Concepts: Activities carried out during design, Classification of Design Methodologies: Function-Oriented, Object-Oriented, Aspect-oriented, Component-based, Properties of a Good Design, Layered Design, Modularity

Function Oriented Design: Overview of SA/SD Methodology, Data Flow Diagrams (DFDs), DFD Model of a System, DFD Model to Structure Chart

Object-Oriented Design: Objects and object classes, An Object-Oriented design process, UML.

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.

Risk management: Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM.

Quality Management: Quality concepts, Software quality assurance, Software reliability, The ISO 9000 quality standards.

Recommended books:

1. Fundamentals of Software Engineering, Rajib Mall, Fourth Edition, PHI, 2016.
2. Software Engineering- Sommerville, 7th edition, Pearson education.
3. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
4. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition. McGrawHill International Edition.

Course Code:	**ITOE-002**
Subject Name	**WEB DESIGN CONCEPTS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Introduction and brief history of world wide web (WWW).
- Web essentials: HTML, XHTML, CSS.
- Addressing web standards, audience requirements and principles of web page design.
- Introduction of Web architecture, databases, jdbc

Course Outcomes

5. Understand basic principles of web site design, considering the information architecture.
6. Incorporate best practices in navigation, usability in website design
7. Design of website adhering to current web standards (HTML, XML, CSS)
8. Learning various scripting languages to create interactive components in web pages.

Course Outcomes	**Program outcomes**											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2
ITOE-002												
CO 1	H		M		M							
CO 2	M	M	H	M	M				H			
CO 3					H	M						
CO 4		H		H		H		H		M		M

Course Content

Introduction to HTML: HTML Common tags- List, Tables, images, forms, Frames; Cascading Style sheets;

Introduction to JavaScript: Scripts, Objects in Java Script, Dynamic HTML with Java Script

XML: Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX

Java Beans: Introduction to Java Beans, Advantages of Java Beans, BDK Introspection, Using Bound properties, Bean Info Interface, Constrained properties Persistence, Customizes, Java Beans API, Introduction to EJB's

Web Servers and Servlets: Tomcat web server, Introduction to Servelets: Lifecycle of a Serverlet, JSDK, The Servelet API, The javax.servelet Package, Reading Servelet parameters, Reading Initialization parameters. The javax.servelet HTTP package, Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues,

Introduction to JSP: The Problem with Servelet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC Setting Up and JSP Environment: Installing the Java Software Development Kit, Tomcat Server & Testing Tomcat

JSP Application Development: Generating Dynamic Content, Using Scripting Elements Implicit JSP Objects, Conditional Processing – Displaying Values Using an Expression to Set an Attribute, Declaring Variables and Methods Error Handling and Debugging Sharing Data Between JSP pages, Requests, and Users Passing Control and Date between Pages – Sharing Session and Application Data – Memory Usage Considerations

Database Access : Database Programming using JDBC, Studying Javax.sql.* package, Accessing a Database from a JSP Page, Application – Specific Database Actions, Deploying JAVA Beans in a JSP Page, Introduction to struts framework.

Recommended books:

1. Chris Bates ,"Web Programming, building internet applications", 3rd Ed., WILEY Dreamtech, 2007.
2. Patrick Naughton and Herbert Schildt, "The complete Reference Java 2" 5th Ed., TMH, 2002.
3. Hans Bergsten , "java Server Pages", 2nd Ed., O'Reilly, 2002.
4. Dietel, Nieto, "Internet and World Wide Web – How to program", 2nd Ed., PHI, 2001.
5. Joel Sklar, "Web Warrier guide to web design technologies", 3rd Ed. Cengage Learning, 2005.

Course Code:	**ITOE-003**
Subject Name	**FUNDAMENTALS OF DATA ANALYTICS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course objectives

- Discuss the overall process of how data analytics is applied
- Discuss how data analytics can be used to better address and identify risks
- Demonstrate the power of data analytics using case studies

Course outcomes

1. Understand what Big Data is and why classical data analysis techniques are no longer adequate.
2. Understand the benefits that Big Data can offer to businesses and organizations.
3. Learn conceptually how Big Data is stored.
4. See how Big Data can be analyzed to extract knowledge.

Course Outcomes	Program outcomes											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITOE-003												
CO 1	M	H	L				L				H	H
CO 2	H	L	M			M			H		M	H
CO 3	M	L	M				L				M	H
CO 4	H	M	H		L				L		M	H

Course content

Introduction: Big Data Overview, Challenges with Big Data, Traditional Business Intelligence (BI) versus Big Data.

Big Data Analytics: The rising and importance of data sciences, Classification of Analytics, Data Sciences, Terminologies in Big Data Environment (In –Memory Analytics, In-Database Processing, Symmetric Multiprocessor system, Massively parallel processing, Parallel and distributed system, Shared nothing architecture, CAP Theorem), Analytics tools.

Hadoop Overview: Introduction to Hadoop, RDBMS vs Hadoop, key aspects of hadoop, hadoop components, hadoop conceptual layer, high level architecture of hadoop.

Hadoop Architecture: Hadoop architecture, Hadoop ecosystem components, Hadoop storage: HDFS, Hadoop processing, Map Reduce Framework, Hadoop server roles.

Hadoop big data technology landscape: NoSQL, Types of NoSQL database, Advantages, New SQL, Comparison of SQL, NoSQL and NewSQL.

Overview of Data Analytics- Theory and Methods: Measures and evaluation, Supervised Learning, Linear and Logistic Regression, Decision trees, Unstructured data analytics.

Recommended Books

4. Big data and Analytics by Seema Acharya and Subhashini Chellappan.
5. Hadoop: The Definitive Guide by Tom White.
6. Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL and Graph by David Loshin.

Course Code:	**ITOE-004**
Subject Name	**AGILE SOFTWARE ENGINEERING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- To explain how an iterative, incremental development process leads to faster delivery of more useful software
- To discuss the essence of agile development methods
- To explain the principles and practices of extreme programming
- To explain the roles of prototyping in the software process

Course Outcomes

1. Describe two or more agile software development methodologies.
2. Identify the benefits and pitfalls of transitioning to agile.
3. Compare agile software development to traditional software development models.
4. Apply agile practices such as test-driven development, standup meetings, and pair programming to their software engineering practices.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITOE-004												
CO 1	H						M		H			
CO 2	M	M	H					H		L		
CO 3	M	H	M					M	M	M		H
CO 4	M	H			M					M	H	M

Course Contents

Overview of Software Engineering: Process, Project, Product, Method, Tool, Modern Life cycle, Traditional development approaches

Advanced Process models: V-Model, Component based development model, Agile Development Model, Unified Process Model, Extreme Programming, Feature Driven development, Lean Software Development, Service Oriented Architecture, Aspect Oriented Development

Agile Project Management: Agile Scrum Framework, Project Planning, Scheduling, Agile Estimation, Iterative Planning, Roles

Software Specification: New paradigms in software specification and design, Agile Specification, Short review of UML.

Design Engineering: Software architecture, Object-oriented Design, Software Patterns, Pattern-oriented Design, Component-oriented design. Software Frameworks, Agile Design

Agile Testing and Test Driven Development: The Agile lifecycle and its impact on testing, Test-Driven Development (TDD), xUnit framework and tools for TDD, Testing user stories - acceptance tests and scenarios, Planning and managing testing cycle, Exploratory testing, Risk based testing, Regression tests, Test Automation, Tools to support the Agile tester

BOOKS RECOMMENDED:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 7th edition. McGraw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson education.
3. Agile Software Development with Scrum By Ken Schawber, Mike Beedle, Pearson, 21 Mar 2008
4. Agile Software Development, Principles, Patterns and Practices By Robert C. Martin, Prentice Hall, 25 Oct 2002

Course Code:	**ITOE-005**
Subject Name	**MOBILE APPLICATION DEVELOPMENT CONCEPTS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- Introduction and characteristics of mobile applications.
- Application models of mobile application frameworks.
- Managing application data and User-interface design for mobile applications.
- Integrating networking, the OS and hardware into mobile-applications.
- Addressing enterprise requirements in mobile applications – performance, scalability, modifiability, availability and security.

Course Outcomes

1. Understand technology and business trends impacting mobile applications
2. Be competent with the characterization and architecture of mobile applications.
3. Understand enterprise scale requirements of mobile applications.
4. Design and develop mobile applications using one application development framework.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITOE-005												
CO 1			M				L					
CO 2		L				M						
CO 3				M				H	H			
CO 4		L					L					

Course content

Introduction: Mobile operating system, Operating system structure, Constraints and Restrictions, Hardware configuration with mobile operating system, Features: Multitasking Scheduling, Memory Allocation, FileSystem Interface, Keypad Interface, I/O Interface, Protection and Security, Multimedia features.

J2ME Overview: Java 2 Micro Edition and the World of Java, Inside J2ME, J2ME and Wireless Devices Small Computing Technology: Wireless Technology, Radio Data Networks, Microwave Technology, Mobile Radio Networks, Messaging, Personal Digital Assistants.

J2ME Architecture and Development Environment: J2ME Architecture, Small Computing Device Requirements, Run-Time Environment, MIDlet Programming, Java Language for J2ME, J2ME Software Development Kits, Hello World J2MEStyle, Multiple MIDlets in a MIDlet Suite, J2ME Wireless Toolkit

J2ME Best Practices and Patterns: The Reality of Working in a J2ME World, Best Practices. Commands, Items, and Event Processing: J2ME User Interfaces, Display Class, The Palm OS Emulator, Command

Class, Item Class, Exception Handling

High-Level Display: Screens: Screen Class, Alert Class, Form Class, Item Class, List Class, TextBox Class, Ticker Class

Low-Level Display: Canvas: The Canvas, User Interactions, Graphics, Clipping Regions, Animation

Record Management System: Record Storage, Writing and Reading Records, Record Enumeration, Sorting Records, Searching Records, Record Listener.

JDBC Objects: The Concept of JDBC, JDBC Driver Types, JDBC Packages, Overview of the JDBC Process, Database Connection, statement Objects, Result set, Transaction Processing, Metadata, Data Types, Exceptions.

JDBC and Embedded SQL: Model Programs, Tables, Indexing, Inserting Data into Tables, Selecting Data from a

Table, Metadata, Updating Tables, Deleting Data form a Table, Joining Tables, Calculating Data, Grouping and

Ordering Data, Subqueries, VIEWs

Android: Introduction to Android, Architecture, memory management, communication protocols, application development methods, deployment, dealing with screen configurations and multiple screen sizes, Activity Class and its lifecycle, intents and permissions, Fragment Class, creating user interfaces, notifying users about important events, handling concurrency, acquiring data over the network, Mobile Cloud Computing with Android (MoCCA).

Recommended books:

6. Jeff McWherter, Scott Gowell “Professional Mobile Application Development”, John Wiley & Sons.
7. J2ME: The Complete Reference, James Keogh, Tata McGraw Hill.
8. Enterprise J2ME: Developing Mobile Java Applications – Michael Juntao Yuan, PearsonEducation.
9. Beginning J2ME: From Novice to Professional, Third Edition, Sing Li, Jonathan B. Knudsen, A press,2005.
10. Jochen Schiller,“Mobile Communications”, Addison-Wesley, 2ndedition, 2004.

Course Code:	**ITOE-006**
Subject Name	**FUNDAMENTALS OF CLOUD COMPUTING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

- To study how to use Cloud Services.
- To implement Virtualization
- To implement Task Scheduling algorithms.
- Apply Map-Reduce concept to applications.
- To build Private Cloud.
- Broadly educate to know the impact of engineering on legal and societal issues involved

Course Outcomes

1. To understand and apply the knowledge of systems protocols and mechanisms to support Cloud computing.
2. To analyze and implement the concepts of Cloud Computing.
3. To deploy applications on cloud middleware.
4. To understand and resolve security issues in various applications.

Course Outcomes	**Program outcomes**											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
ITOE-006												
CO 1	L	H		M								
CO 2	L	M	M					M				
CO 3	L	M	M			H				H		
CO 4	L	M	M		M							

Course content

Cloud Computing, History of Cloud Computing, Cloud Architecture, Cloud Storage, Why Cloud Computing Matters, Advantages of Cloud Computing, Disadvantages of Cloud Computing, Companies in the Cloud Today, Cloud Services

Web-Based Application, Pros and Cons of Cloud Service Development, Types of Cloud Service Development, Software as a Service, Platform as a Service, Web Services, On-Demand Computing, Discovering Cloud Services Development Services and Tools, Amazon Ec2, Google App Engine, IBM Clouds

Centralizing Email Communications, Collaborating on Schedules, Collaborating on To-Do Lists, Collaborating Contact Lists, Cloud Computing for the Community, Collaborating on Group Projects and Events, Cloud Computing for the Corporation

Collaborating on Calendars, Schedules and Task Management, Exploring Online Scheduling Applications, Exploring Online Planning and Task Management, Collaborating on Event Management, Collaborating on Contact Management, Collaborating on Project Management, Collaborating on Word Processing - Collaborating on Databases, Storing and Sharing Files

Collaborating via Web-Based Communication Tools, Evaluating Web Mail Services, Evaluating Web

Conference Tools, Collaborating via Social Networks and Groupware, Collaborating via Blogs and Wikis

Recommended books:

1. Michael Miller, Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, Que Publishing, August 2008.
2. Haley Beard, Cloud Computing Best Practices for Managing and Measuring Processes for On-demand Computing, Applications and Data Centers in the Cloud with SLAs, Emereo Pty Limited, July 2008.
3. Thomas Erl and Ricardo Puttini, "Cloud Computing : Concepts Technology & Architecture" , PHI.
4. Kailash Jayaswal and Jagannath Kallakurchi , "Cloud Computing Black Book" , Willey India.
5. Buyya and Vecchiola , "Mastering Cloud Computing" , Morgan Kaufmann *Publishers.*

Course Code:	**ITOE-007**
Subject Name	**MACHINE LEARNING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

This course will serve as a comprehensive introduction to various topics in machine learning. At the end of the course the students should be able to design and implement machine learning solutions to classification, regression, and clustering problems; and be able to evaluate and interpret the results of the algorithms.

Course Outcomes

1. Understanding about the basic principles, techniques and applications of machine learning.
2. Broad understanding of machine learning algorithms and their use in data-driven knowledge discovery and program synthesis.
3. Knowledge of the strengths and weaknesses of different machine learning algorithms (relative to the characteristics of the application domain).
4. The ability to adapt or combine some of the key elements of existing machine learning algorithms to design new algorithms as needed.

Course Outcomes	**Program outcomes**											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPC-308												
CO 1		M	M		M			M			H	
CO 2		M	H	M	M				H	H	M	
CO 3					H	M			H		M	
CO 4		H		H		M		M		M		M

Course Content
Introduction: Well-Posed Learning Problems, Designing a Leaning System, Perspectives and Issues in Machine Learning.

Concept Learning and the General-to-Specific Ordering: Introduction, A Concept Learning Task, Concept Learning as Search, FIND-S: Finding a Maximally Specific Hypothesis, Version Spaces and the CANDIDATE-ELIMINATION Algorithm.

Decision Tree Learning: Introduction, Decision Tree Representation, Appropriate problem for Decision tree Learning, The Basic Decision Tree Learning Algorithm, Hypothesis Space Search in Decision Tree Leaning, Inductive Bias in Decision Tree Leaning, Issues in Decision Tree Leaning

Artificial Neural Networks: Introduction, Natural Network Representations, Appropriate Problems for Neural Network Learning, Perceptions, Multilayer Network and the BACKPROPAGATION Algorithm

Bayesian Learning: Introduction, Bayes Theorem, Bayes Theorem and Concept Learning, Bayes Optimal Classifier, Native Bayes Classifier, An Example: Learning to Classify Text.

Instance- Based Learning: Introduction, K-NEAREST NEIGHBOUR Learning, Distance-Weighted NEAREST NEIGHBOUR Algorithm.

Genetic Algorithms: Motivation, Genetic Algorithms, Hypothesis Space Search, Genetic Programming, Parallelizing Genetic Algorithms.

Learning Sets of Rules: Introduction, Sequential Covering Algorithms, Learning Rule Sets: Summary, Learning First-Order Rules, Learning Sets of First-Order Rules: FOIL, Induction as Inverted Deduction, Inverted Resolution.

Support Vector Machine: Maximum margin linear separators, Quadratic Programming Solution to finding maximum margin separators, Kernels for learning non-linear functions.

Recommended Books

5. Tom Mitchell, “Machine Learning”, McGraw-Hill, 1997.
6. E. Alpaydin, “Machine Learning”, MIT Press, 2010.
7. K. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
8. C. Bishop, “Pattern Recognition and Machine Learning, Springer”, 2006.

Course Code:	**ITOE-008**
Subject Name	**Fundamentals of Artificial Intelligence**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course objectives

- To provide a strong foundation of fundamental concepts in Artificial Intelligence
- To provide a basic exposition to the goals and methods of Artificial Intelligence
- To enable the student to apply these techniques in applications which involve perception, reasoning and learning

Course Outcomes

5. Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.
6. Formalize a given problem in the language/framework of different AI methods
7. Describe basic AI algorithms (e.g., standard search algorithms or resolution).
8. Design and carry out an empirical evaluation of different algorithms on a problem formalization.

Course Outcomes	Program outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ITPE-063												
CO 1	H	M				L		L				
CO 2			M	H	H							
CO 3		M		M	L				M			
CO 4	H				H							H

Course Content

Introduction: Introduction to AI: Definitions, Historical foundations, Basic Elements of AI, Characteristics of intelligent algorithm, AI application Areas, Introduction to genetic algorithm, Evolutionary searching algorithm.

Problem solving: State space search; Production systems, search space control: depth-first, breadth-first search, heuristic search - Hill climbing, best-first search, branch and bound. Problem Reduction, Constraint Satisfaction End, Means-End Analysis

Handling uncertainty: Non-Monotonic Reasoning, Probabilistic reasoning, use of certainty factors, Fuzzy logic

Knowledge Based Systems: Proportional Logic, FOPL, Clausal Form, Resolution & Unification. Knowledge representation, acquisition, organisation & Manipulation, Semantic nets, Frames, Conceptual Dependency, Scripts & CYC.

Machine Learning: Concept of learning, Concept creation, learning automation, supervised and Unsupervised Learning, learning tasks & learning strategies, single layer & multiplayer Perceptions, Back propagation, learning by inductions, Competitive Learning, Hebbian Coincidence Learning, Attractor Networks Samuel's checkers algorithm. Hopfield nets, Adaptive resonance theory

Expert Systems: Need and justification for expert systems, Basic Components & architecture of Expert systems, ES-Shells, Representing & Using Domain Knowledge, Knowledge acquisition in expert Systems. Case studies: MYCIN, RI.

Recommended Books

7. Rich and K. Knight,"Artificial Intelligence", Tata McGraw Hill.
8. George F. Luger, “Artificial Intelligence – Structures and Strategies for Complex Problem Solving", Pearson Education.
9. Russell &Norvig, "Artificial Intelligence ‘a Modern Approach", Pearson Education.
10. Dan W. Patterson, “Introduction to Artificial Intelligence and Expert Systems”, PHI.
11. E. Charnaik and D. McDermott," Introduction to Artificial Intelligence", Addison-Wesley Publishing Company.
12. Nils J. Nilson, “Principles of Artificial Intelligence”, Narosa Publishing Co.