

**GANGADHAR MEHER UNIVERSITY**  
**AMRUTA VIHAR, SAMBALPUR, ODISHA**



**SCHOOL OF COMPUTER SCIENCE**

*Syllabus for*  
**M.Sc. in Computer Science**

**(2-Year Programme)**

**Course Effective from Academic year 2020-2021**

## Post Graduate Programme Structure

| Year        | Semesters    |             |
|-------------|--------------|-------------|
| First Year  | Semester I   | Semester II |
| Second Year | Semester III | Semester IV |

### Part-I: Semester-I

| Papers       |                                  | Marks    |          | Total Marks | Duration (Hrs) | Credit    |
|--------------|----------------------------------|----------|----------|-------------|----------------|-----------|
| Paper No     | Title                            | Mid Term | End Term |             |                |           |
| CSC101       | Discrete Mathematical Structures | 20       | 80       | 100         | 4              | 4         |
| CSC102       | Computer Architecture            | 20       | 80       | 100         | 4              | 4         |
| CSC103       | Data Communication & Networks    | 20       | 80       | 100         | 4              | 4         |
| CSC104       | Advanced Data Structures         | 20       | 80       | 100         | 4              | 4         |
| CSC105       | Lab (Data Structure & Python)    | 20       | 80       | 100         | 4              | 4         |
| <b>Total</b> |                                  |          |          | <b>500</b>  |                | <b>20</b> |

### Part-I: Semester-II

| Papers              |   | Marks    |          | Total Marks | Duration (Hrs) | Credit    |
|---------------------|---|----------|----------|-------------|----------------|-----------|
| Paper No            | Title                                     | Mid Term | End Term |             |                |           |
| CSC201              | Object Oriented Programming using JAVA    | 20       | 80       | 100         | 4              | 4         |
| CSC202              | Mobile Computing                          | 20       | 80       | 100         | 4              | 4         |
| CSC203              | Advanced Operating Systems                | 20       | 80       | 100         | 4              | 4         |
| CSC204              | Theory of Computation                     | 20       | 80       | 100         | 4              | 4         |
| CSC205              | Lab (Java and OS)                         | 20       | 80       | 100         | 4              | 4         |
| <b>DSE –IPapers</b> |   |          |          |             |                |           |
| CSC206 A            | <b>Data Warehousing and Mining</b>        | 20       | 80       | 100         | 4              | 4         |
| CSC206 B            | <b>Wireless Sensor Networks</b>           | 20       | 80       | 100         | 4              | 4         |
| CSC206 C            | <b>Internet of Things</b>                 | 20       | 80       | 100         | 4              | 4         |
| CSC206D             | <b>Microprocessor and Microcontroller</b> | 20       | 80       | 100         | 4              | 4         |
| <b>Total</b>        |   |          |          | <b>600</b>  |                | <b>24</b> |

**Part-II: Semester-III**

| Papers             |   | Marks    |          | Total Marks | Duration (Hrs) | Credit    |
|--------------------|---|----------|----------|-------------|----------------|-----------|
| Paper No           | Title   | Mid Term | End Term |             |                |           |
| CSC301             | Compiler Construction                           | 20       | 80       | 100         | 4              | 4         |
| CSC302             | Data Base Management Systems                    | 20       | 80       | 100         | 4              | 4         |
| CSC303             | Design and Analysis of Algorithms               | 20       | 80       | 100         | 4              | 4         |
| CSC304             | Web Technology                                  | 20       | 80       | 100         | 4              | 4         |
| CSC305             | Lab (Web Technology and DBMS)                   | 20       | 80       | 100         | 4              | 4         |
| <b>IDSE Papers</b> |   |          |          |             |                |           |
| CSC306A            | <b>Network and Internet Technologies</b>        | 20       | 80       | 100         | 4              | 4         |
| CSC306B            | <b>Fundamentals of Computer</b>                 | 20       | 80       | 100         | 4              | 4         |
| CSC306C            | <b>Introduction to Programming Using Python</b> | 20       | 80       | 100         | 4              | 4         |
| <b>Total</b>       |   |          |          | <b>600</b>  |                | <b>24</b> |

**Part-II: Semester-IV**

| Papers           |                                   | Marks    |          | Total Marks | Duration (Hrs) | Credit    |
|------------------|-----------------------------------|----------|----------|-------------|----------------|-----------|
| Paper No         | Title                             | Mid Term | End Term |             |                |           |
| CSC401           | Machine Learning                  | 20       | 80       | 100         | 4              | 4         |
| CSC402           | Software Engineering & OOAD       | 20       | 80       | 100         | 4              | 4         |
| CSC403           | Information Security              | 20       | 80       | 100         | 4              | 4         |
| CSC404           | Cloud Computing                   | 20       | 80       | 100         | 4              | 4         |
| CSC405           | Project Work Report and VIVA VOCE |          |          | 100         |                | 4         |
| <b>Total</b>     |                                   |          |          | <b>500</b>  |                | <b>20</b> |
| <b>22 Papers</b> | <b>Grand Total</b>                |          |          | <b>2200</b> |                | <b>88</b> |

| <b>SEMESTER WISE CREDIT DISTRIBUTION</b> |           |           |            |           |              |
|--|-----------|-----------|------------|-----------|--------------|
| <b>Semester</b>                          | <b>I</b>  | <b>II</b> | <b>III</b> | <b>IV</b> | <b>TOTAL</b> |
| <b>Total Credit</b>                      | <b>20</b> | <b>24</b> | <b>24</b>  | <b>20</b> | <b>88</b>    |

## Semester-I

Course Name:

### DISCRETE MATHEMATICAL STRUCTURES

Category:

Programme Core Course

Prerequisite:

Basics of set theory and combinatory.

Learning Objective:

- The objective is to introduce Logic, Graphs and Algebraic structures.

Learning Outcome:

- Logics and graphs are the key points for algorithm, Networking, coding and many more recent areas. This course helps to understand some areas of computer science in detail.

|   |
|---|
| <b>Paper-CSC101</b>   |
| <b>Discrete Mathematical Structures</b>   |
| <b>UNIT-I:</b>  |
| <b>Fundamentals of Logic:</b> Propositional Logic, Propositional Equivalences, Predicate and Quantifiers, nested Quantifiers, Rules of Inference. <b>Set Theory:</b> Sets, Set Operations. <b>Introduction to proofs:</b> proof by Induction, proof by contradiction, proof by cases with examples. <b>Mathematical Induction:</b> Introduction to Induction, strong Induction, Recursion. <b>Relations:</b> Relations and their properties, n-ary Relations and their applications, Representing relations, Closures of relations, Equivalence relations, and Partial Orderings. |
| <b>UNIT-II:</b>   |
| <b>Number Theory:</b> The division algorithm, Remainders, greatest common divisors, The fundamental theorem of arithmetic, infinity of primes. <b>Graphs:</b> Graphs, Graph models, special types of graphs, Representing graphs, Graph Isomorphism, connectivity, Euler and Hamilton paths, Planar graphs, Graph Coloring, Matching problem. <b>Trees:</b> Introduction to Trees, Applications of Trees, Binary Trees, n-ary Trees, Tree Traversal, Spanning Trees.  |
| <b>UNIT-III:</b>  |
| <b>Principles of Counting:</b> Counting using Sum Rule and Product rule. Concepts of permutations, combinations and circular permutations. Pigeonhole Principle, Occupancy Problem. <b>Advanced Counting Techniques:</b> Recurrence relations, solving linear recurrence relations, Generating functions, Catalan Numbers, Principle of Inclusion and Exclusion, Applications of Inclusion and Exclusion.   |
| <b>UNIT-IV:</b>   |
| <b>Basic Probability:</b> Random experiment. Sample space. Mutually exclusive events. Empirical definition of probability. Problems based on probability. Axiomatic definition of probability. Properties based on axiomatic definition of probability. Conditional probability. Independent events. <b>Bayes' Theorem and Applications:</b> Bayes' Theorem and problems based on conditional probability.  |
| <b>Text Book:</b>   |
| 1. K. H. Rosen, <b>Discrete Mathematics &amp; Its Applications( with Combinatorics and Graph Theory)</b> , 6 <sup>th</sup> Edition, McGraw-Hill, 2007.  |
| <b>Reference Books:</b>   |

1. C.L.Liu, D. P. Mohapatra, **Elements of Discrete Mathematics: A Computer Oriented Approach**, 3<sup>rd</sup> Edition, McGraw-Hill, 2008.
2. J.P. Tremblay, R. Manohar, **Discrete Mathematical Structures with Applications to Computer Science**, McGraw-Hill, 1997.
3. R. L. Graham, D. E. Knuth, O. Patashnik, **Concrete Mathematics : A Foundation for Computer Science**, 2<sup>nd</sup> Edition, Pearson Education, 2007.
4. D. B. West, **Introduction to Graph Theory**, 2<sup>nd</sup> Edition, PHI Learning, 2009.
5. R. A. Brualdi, **Introductory Combinatorics**, 4<sup>th</sup> Edition, Pearson, 2004.

**Course Name:**

# COMPUTER ARCHITECTURE

**Category:**

Programme Core Course

**Prerequisite:**

Digital Logic

**Learning Objective:**

- To study design of an elementary basic computer
- To have a better understanding of a hardwired and microprogrammed control unit.
- To introduce the concept of memory hierarchy and pipelining to speed-up the processor

**Learning Outcome:**

- After this course students understand in a better way to design and interconnection of various modules of a system, the I/O and memory organization in depth.

|   |
|---|
| <b>Paper-CSC102</b>   |
| <b>Computer Architecture</b>  |
| <b>UNIT-I:</b>  |
| <b>Register Transfer and Micro-operations:</b><br>Register Transfer Language, Register transfer, Bus and memory transfer, Arithmetic, Logical and Shift Micro Operation, Arithmetic Logic Shift Unit<br><b>Basic Computer Organization and Design:</b><br>Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle & Register Reference Instructions, Memory Reference Instructions, Input-Output and Interrupt. Design of Basic Computer.   |
| <b>UNIT-II:</b>   |
| <b>Basic Processing Unit:</b><br>Some Fundamental Concepts: Register Transfers, Performing an Arithmetic or Logic Operation, Fetching a Word from Memory, Storing a word in Memory, Execution of Complete Instruction, Branch Instructions, Multiple Bus Organization<br><b>Micro-Programmed Control:</b><br>Control Memory, Address Sequencing: Conditional branching, Mapping of Instruction, Subroutine; Micro Program Example: Computer configuration, Microinstruction format, Symbolic Microinstruction, The fetch routine, Symbolic Microprogram, Binary Microinstruction; Design of Control Unit: Microprogram Sequencer. |
| <b>UNIT-III:</b>  |
| <b>Memory Organization:</b><br>Memory Hierarchy, Associative Memory: Hardware Organisation, Match Logic, Read Operation, Write Operation; Cache Memory: Associative Mapping, Direct Mapping, Set Associative Mapping, Write into Cache Memory, Cache Initialization; Virtual Memory: Address Space and Memory Space, Address mapping Using Pages, Associative Memory Page Table, Page Replacement   |
| <b>UNIT-IV:</b>   |
| <b>Pipeline and Vector Processing:</b><br>Parallel Processing, Pipelining: General Considerations; Arithmetic Pipeline, Instruction Pipeline: Four-segment Instruction Pipeline Example, Data Dependency, Handling of Branch Instructions; Vector Processing; Array Processors<br><b>Multiprocessors:</b>   |

Characteristics of Multiprocessors, Interconnection structures, Interprocessor arbitration, Interprocessor communication and synchronization, Cache coherence

**Text Books:**

1. Computer Organization & Architecture-William Stallings, 8<sup>th</sup> Edition (PHI)

**Reference Books:**

1. Computer Architecture and Organization- Rajiv Chopra (S. Chand)
2. Computer Organization- Carl Hamacher, Zvonko Vranesic, Safwat Zaky 5th Edition, McGraw-Hill Education India

**Course Name:** DATA COMMUNICATION & NETWORKS

**Category:** Programme Core Course

**Prerequisite:** Basics of Computer

**Learning Objective:**

- The objective of the course is to provide an overview of communication network functions and a good foundation for further studies in the subject. It involves understanding and application of design principles and methods for systems development and review of the underlying systems, and communications technologies and significant standardized systems.

**Learning Outcome:**

- Understand and be able to explain the principles of layered protocol architecture; be able to identify and describe the system functions in the correct protocol layer and further describe how the layers interact.

|   |
|---|
| <b>Paper-CSC103</b>   |
| <b>Data Communication &amp; Networks</b>  |
| <b>UNIT-I:</b>  |
| Overview of Data Communications and Networking. <b>Physical Layer:</b> Analog and Digital, Analog Signals, Digital Signals, Analog versus Digital, Data Rate Limits, Transmission Impairment, More about signals. Digital Transmission: Line coding, Block coding, Sampling, Transmission mode. Analog Transmission: Modulation of Digital Data; Telephone modems, modulation of Analog signals. Multiplexing: FDM , WDM , TDM , Transmission Media: Guided Media, Unguided media (wireless) Circuit switching and Telephone Network: Circuit switching, Telephone network. |
| <b>UNIT-II:</b>   |
| <b>Data Link Layer:</b> Error Detection and correction: Types of Errors, Detection, Error Correction Data Link Control and Protocols: Flow and Error Control, Stop-and-wait ARQ. Go-Back-N ARQ, Selective Repeat ARQ, HDLC, Point –to- Point Protocol, Multiple Access, Random Access, Controlled Access, Channelization. Local area Network: Ethernet, Traditional Ethernet, Fast Ethernet, Gigabit Ethernet. Wireless LANs: IEEE 802.11, Bluetooth virtual circuits: Frame Relay and ATM.   |
| <b>UNIT-III:</b>  |
| <b>Network Layer:</b> Host to Host Delivery: Internetworking, addressing, Routing. Network Layer Protocols: ARP, RARP, NAT, BOOTP, DHCP, IPV4, ICMP, IPV6, ICMPV6 and Unicast routing protocols Transport Layer: Process to Process Delivery: UDP, TCP, congestion control and Quality of service.  |
| <b>UNIT-IV:</b>   |
| <b>Application Layer:</b> Client Server Model, Peer to peer network, Domain Name System (DNS): Electronic Mail (SMTP) and file transfer (FTP) HTTP and WWW.   |
| <b>Text Book:</b>   |
| 1. B. A. Forouzan, <b>Data Communications and Networking</b> , 4 <sup>th</sup> Edition, Tata McGraw-Hill 2007.  |
| <b>Reference Books:</b>   |



1. Computer Networks -A S Tenenbaum (PHI)

2. Computer Networking: A Top-Down Approach Featuring the Internet, 3/e - James F. Kurose & Keith W. Ross (Pearson Education India)

**Course Name:** **ADVANCED DATA STRUCTURES**

**Category:** Programme Core Course

**Prerequisite:** **Programming in C**

**Learning Objective:**

- To get clear understanding about the basic data structures and their operations, the concepts of algorithms, basic search and sort algorithms.
- Student will also gain adequate knowledge to choose appropriate data structure and algorithm to solve a problem.

**Learning Outcome:**

- Students should be able to select an appropriate data structure in the solution for a real world problem.

|   |
|---|
| <b>Paper-CSC104</b>   |
| <b>Advanced Data Structures</b>   |
| <b>UNIT-I:</b><br>Introduction to Data Structures, Arrays and Strings, Introduction to Algorithms, Algorithm development, Complexity analysis, Recursion.<br><b>Linear Data Structures:</b> Stacks: Operations and Applications, Queues: Operations and Applications, Circular Queues: Operations and Applications.<br><b>Linked Lists:</b> Operation – Creations, insertion, Deletion, Circular Lists, and Doubly Linked List.                         |
| <b>UNIT-II:</b><br><b>Sorting:</b> Insertion Sort, Merge Sort, Quick Sort, Radix Sort, and Heap Sort.<br><b>Searching:</b> Binary Search, Selection.<br><b>Dictionaries:</b> skip-lists, hashing, analysis of collision resolution techniques.  |
| <b>UNIT-III:</b><br><b>Search Trees-</b> Binary search Trees, Threaded binary tree, AVL Trees, B Trees, Red Black trees. Searching, insertion, deletion operations of trees.<br><b>Tries and pattern matching:</b> Priority queues and binary heaps   |
| <b>UNIT-IV:</b><br><b>Introduction to Graphs:</b> Breadth first search and connected components. Depth first search in directed and undirected graphs and strongly connected components.<br><b>Spanning trees:</b> Prim's and Kruskal's algorithm, union-find data structure. Dijkstra's algorithm for shortest path. Shortest path tree. Shortest and longest paths in directed acyclic graphs. Automatic List management, dynamic storage management. |
| <b>Text Book:</b><br>1. Y. Langsam, M. Augenstein, A. M. Tenenbaum, <b>Data Structure using C and C++</b> , Prentice Hall, 1996.  |
| <b>Reference Books:</b><br>1. E. Horowitz, D. Mehta, S. Sahani. <b>Fundamentals of Data Structures in C++</b> , Universities Press. 2007.<br>2. M. A. Weiss, <b>Data Structures and Algorithm Analysis in C++</b> , Pearson Education 2006.<br>3. M. T. Goodrich, R. Tamassia, D. Mount, <b>Data Structures and Algorithms in C++</b> , Wiley India Pvt. Ltd, 2004  |

**Lab ( Data Structures and Python )**

**DATA STRUCTURE PROGRAMS:**

1. Implementation of sparse matrix
2. Implementation of linear search, binary search, bubble sort, insertion sort, selection sort
3. Implementation of single linked list and its operations
4. Design a doubly linked list to hold strings and use it for organizing a sequence of cities
5. Repeat Q4 using doubly circular linked list
6. Create a polynomial using single linked list and perform addition operation of two polynomials
7. Implement a stack,use stack for conversion of infix to postfix and evaluation of postfix expression.
8. Implementation of circular queue (using array) with menu options like insert, delete,display and exit.
9. Implementation of a priority queue and use it to organize studentrecords prioritized by marks.
10. Recursive implementation of quick sort and merge sort. Generate 10 random integers in a given range and apply sorting mechanisms.
11. Implement linear search and binary search to find out whether a given element is present or not in the array. Compare two search mechanisms based on number of comparisons required for a successful as well as unsuccessful search.
12. Implementation of a binary search tree with menu options: Construct a tree, insertanode, delete anode, traverse and display preorder, in order and post order sequenceof its nodes.
13. Implementation of Heap Sort.
14. Implementation of digraphs using adjacency matrix and find the transitive closureusingWarshall's algorithm.
15. Implementation of a weighted graph and find minimal cost spanning tree usingPrim's algorithm.
16. Implementation of a weighted graph and find minimal cost spanning tree using Kruskal's algorithm.
17. Implement Dijkstra's algorithm to find single source shortest path.
18. Implement Floyd Warshall's algorithm to find all pair shortest path.
19. Implement Topological sorting.
20. Implementation of a small Real World Application illustrating data structure usage.

**PYTHON LAB:**

**Exercise 1 - Basics**

- a) Running instructions in Interactive interpreter and a Python Script
- b) Write a program to purposefully raise Indentation Error and Correct it

**Exercise 2 - Operations**

- a) Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)
- b) Write a program add.py that takes 2 numbers as command line arguments and prints its sum.

**Exercise - 3 Control Flow**

- a) Write a Program for checking whether the given number is a even number or not.
- b) Using a for loop, write a program that prints out the decimal equivalents of  $1/2, 1/3, 1/4, \dots, 1/10$
- c) Write a program using a for loop that loops over a sequence. What is sequence?
- d) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

#### **Exercise 4 - Control Flow - Continued**

- a) Find the sum of all the primes below two million.  
Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:  
1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...
- b) By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

#### **Exercise - 5 - DS**

- a) Write a program to count the numbers of characters in the string and store them in a dictionary data structure
- b) Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.

#### **Exercise - 6 DS - Continued**

- a) Write a program combine\_lists that combines these lists into a dictionary.
- b) Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?

#### **Exercise - 7 Files**

- a) Write a program to print each line of a file in reverse order.
- b) Write a program to compute the number of characters, words and lines in a file.

#### **Exercise - 8 Functions**

- a) Write a function ball\_collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.  
Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius If (distance between two balls centers)  $\leq$  (sum of their radii) then (they are colliding)
- b) Find mean, median, mode for the given set of numbers in a list.

#### **Exercise - 9 Functions - Continued**

- a) Write a function nearly\_equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.
- b) Write a function dups to find all duplicates in the list.
- c) Write a function unique to find all the unique elements of a list.

#### **Exercise - 10 - Functions - Problem Solving**

- a) Write a function cumulative\_product to compute cumulative product of a list of numbers.
- b) Write a function reverse to reverse a list. Without using the reverse function.
- c) Write function to compute GCD, LCM of two numbers. Each function shouldn't exceed one line.

**Exercise 11 - Multi-D Lists**

- a) Write a program that defines a matrix and prints
- b) Write a program to perform addition of two square matrices
- c) Write a program to perform multiplication of two square matrices

**Exercise - 12 - Modules**

- a) Install packages requests, flask and explore them using (pip)
- b) Write a script that imports requests and fetch content from the page. Eg. (Wiki)
- c) Write a simple script that serves a simple HTTP Response and a simple HTML Page

**Exercise - 13 OOP**

- a) Class variables and instance variable
  - i) Robot
  - ii) ATM Machine

**Exercise - 14 GUI, Graphics**

- 1. Write a GUI for an Expression Calculator using tk.

## Semester-II

**Course Name:** OBJECT ORENATED PROGRAMMING USING JAVA

**Category:** Programme Core Course

**Prerequisite:** Basic procedural programming Language (like C-Programming)

**Learning Objective:**

- Introduces object oriented programming concepts using the Java language.
- Introduces the principles of inheritance and polymorphism; and demonstrates howthey relate to the design of abstract classes.
- Introduces the implementation of packages and interfaces.
- Introduces exception handling, event handling and multithreading.
- Introduces the design of Graphical User Interface using applets and swings.

**Learning Outcome:**

- Develop applications for a range of problems using object-oriented programming techniques
- Design simple Graphical User Interface applications

|   |
|---|
| <b>Paper-CSC201</b>   |
| <b>Object Oriented Programming using JAVA</b>   |
| <b>UNIT-I:</b>  |
| <b>Java Evolution and Environment:</b> Java evolution, overview of java language, java history, features of java, how java differs from C and C++, java and World Wide Web, web browser.<br><b>Java Environment:</b> Java Development Kit(JDK), Application Programming Interface(API), java programming structure, java tokens, constants, variables, expressions, decision making statements and looping, java statements, overview of arrays and strings, machine neutral, Java Virtual Machine(JVM), Command Line Arguments.  |
| <b>UNIT-II:</b>   |
| <b>Classes, Objects and Methods:</b> Introduction, defining a class, creating objects, accessing class members, constructors, method overloading, static members. Inheritance: Defining a sub-class, sub-class constructor, multi-level variables, final classes and finalize methods, abstract methods and classes, visibility control. <b>Arrays and Strings:</b> One-dimensional arrays, creating an array, declaration of arrays, initialization of arrays, two-dimensional arrays, string arrays, string methods, string buffer class, vectors, wrapper classes, Basic I/O Streams: Scanner, buffered reader, Collection classes. <b>Managing Errors and Exceptions:</b> Introduction, types of errors: compile time and run-time errors, exceptions, types of exceptions, syntax of exception handling code, multiple catch statements, using finally statement, throwing our own exceptions. |
| <b>UNIT-III:</b>  |
| <b>Interfaces, Package and Multi-threaded Programming:</b> Introduction, defining interfaces, extended interfaces, implementing interfaces. <b>Package:</b> Creation, importing a package and user-defined package. <b>Threads:</b> Introduction to threads, creating threads, extending the thread class, implementing the 'runnable' interface, life-cycle of a thread, priority of a thread, synchronization, and deadlock.  |
| <b>UNIT-IV:</b>   |
| <b>Applet programming:</b> Introduction, how applets differ from applications, building applet code, applet life cycle, about HTML, designing a web page, passing parameters to applets, getting  |

input from the user.**Graphics Programming:** Introduction, abstract window toolkit class hierarchy, frames, event-driven programming, layout managers, panels, canvases, drawing geometric figures.**Introduction to Swings:** Introduction to Swings, overview of Swing components: JButton, JCheckBox, JRadioButton, JLabel, JTextField, JTextArea, JList.

**Text Book:**

1. H. Schildt, **The Java Complete References**, 9<sup>th</sup> Edition, Tata McGraw Hill, 2014.

**Reference Books:**

1. Y.Daniel Liang, **An Introduction to JAVA Programming**, Tata McGraw Hill, 2009.
2. K. Sierra, **Head First java**, 2<sup>nd</sup> Edition, Shroff Publishers, 2012.
3. 3. E. Balaguruswamy, **Programming with JAVA**, 2/e, Tata McGraw Hill

Course Name:

## MOBILE COMPUTING

Category:

Programme Core Course

Prerequisite:

Data Communication and Computer Networks

Learning Objective:

- Describe wireless and mobile communications systems and be able to choose an appropriate mobile system from a set of requirements.

Learning Outcome:

- Be able to avoid or work around the weaknesses of mobile computing, or to reject mobile computing as a solution.
- Interface a mobile computing system to hardware and networks.
- Program applications on a mobile computing system and interact with servers and database systems.

### Paper-CSC202

## Mobile Computing

### UNIT-I:

Introduction to mobile computing, mobile computing architecture, mobile devices, mobile system networks: Cellular Network and frequency reuse, Channel Assignment, Handoff Strategies, Interferences and System Capacity, Improving coverage and capacity in Cellular Systems – Cell Splitting, Sectoring, Repeaters and Range Extension, Limitations of Mobile Computing.

### UNIT-II:

**Personal Communications Services (PCS):** PCS Architecture, mobility management, Global System for Mobile Communication (GSM). System overview: GSM Architecture, Mobility management, Network signalling. General Packet Radio Services (GPRS): GPRS Architecture, GPRS Network Nodes

### UNIT-III:

**Mobile Network Layer:** Mobile IP (Goals, assumptions, entities and terminology, IP packet delivery, agent advertisement and discovery, registration, tunnelling and encapsulation, optimizations), Dynamic Host Configuration Protocol (DHCP).

**Mobile Transport Layer:** Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP, Fast retransmit/fast recovery, Transmission /time-out freezing, Selective retransmission, Transaction oriented TCP.

### UNIT-IV:

**Mobile Data Communication:** WLANs (Wireless LANs), IEEE 802.11 standards.

**Mobile Satellite Communication Networks:** Case studies of the IRIDIUM and GLOBALSTAR systems. Wireless Enterprise Networks: Introduction to Virtual Networks, VPN, Mobile Ad-hoc networks, 4G Technology, Long Term Evolution (LTE).

### Text Book:

1. R. Kamal, **Mobile Computing**, 1<sup>st</sup> Edition, Oxford University Press, 2006.
2. Mobile Computing Technology, Applications & Service Creation, A K Talukder & R RYavagal (TMH)
3. Wireless Communication, T. S Rappaport, Pearson

### Reference Books:

1. Mobile Communications - Jochen Schiller (Addison-Wesley, Second Edition, 2009)
2. Principles of Mobile Computing - UWE Hansmann, LotherMerk, Martin S. Nicklaus, Thomas Stober (Second Edition, Springer)
3. Third Generation Mobile Telecommunication Systems, by P. Stavronlakis, Springer Publishers



Course Name:

## ADVANCED OPERATING SYSTEMS

Category:

Programme Core Course

Prerequisite:

- Computer Programming and Data Structures
- Computer Organization and Architecture

Learning Objective:

- Provide an introduction to operating system concepts (i.e., processes, threads, scheduling,
- synchronization, deadlocks, memory management, file and I/O subsystems and protection)
- Introduce the issues to be considered in the design and development of operating system

### Learning outcome:

- To master the concepts of a process and how the processes are scheduled and synchronized.
- To develop the understanding of detecting a deadlock situation and be able to recovery from it.
- To understand the different approaches to memory management and disk management.
- To understand the structure and organization of the file systems and I/O systems

|   |
|---|
| <b>Paper-CSC203</b>   |
| <b>Advanced Operating Systems</b>   |
| <b>UNIT-I:</b>  |
| <b>Operating System Overview:</b> -Introduction, The Need of Operating Systems, Evolution of Operating Systems, Types of Operating Systems, Simple Batch, Multiprogrammed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls, Virtual Machines, System Design and Implementation.<br><b>Process Management</b> – Process concepts, Life cycle, PCB, Schedulers, Process Scheduling, Threads, Scheduling Levels, CPU Scheduling: Scheduling-Criteria, Algorithms, Algorithm Evaluation. |
| <b>UNIT-II:</b>   |
| <b>Concurrency:</b> -Process synchronization, The Critical- Section Problem, Peterson’s Solution, synchronization Hardware, Semaphores, Classic problems of synchronization, Monitors.<br><b>Deadlocks:</b> System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock detection, deadlock prevention, deadlock avoidance, Recovery from deadlock.   |
| <b>UNIT-III:</b>  |
| <b>Memory Management:</b> Main Memory, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table.<br><b>Virtual Memory:</b> Demand Paging, Page Replacement, Allocation of Frames, Thrashing, Memory-Mapped Files   |
| <b>UNIT-IV:</b>   |
| <b>Mass-Storage Structure:</b> Overview, Disk Structure, Disk scheduling, disk management, Swap-space management, RAID structure.   |

**File Systems:** File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, Protection. File- System Structure and Implementation, Directory Implementation, Allocation Methods, Free-Space Management.

**Text Book:**

1. A. Silberchatz, P. B. Galvin, G. Gagne, **Operating System Concepts**, 7th Edition, John Wiley, 2003.

**Reference Books:**

1. Charles Crowley, **Operating systems : A design-oriented approach**, McGraw-Hill, 1996.
2. A. S. Tanenbaum and H. Bos, **Modern Operating Systems**, Pearson, 2015.
3. W. Stallings, **Operating Systems – Internals and Design Principles**, Prentice Hall, 2009.
4. D. M. Dhandhere, **Operating Systems-A Concept Based Approach**, McGraw-Hill, 2006.

|                            |   |
|----------------------------|---|
| <b>Course Name:</b>        | <b>THEORY OF COMPUTATION</b>  |
| <b>Category:</b>           | Programme Core Course   |
| <b>Prerequisite:</b>       | <b>Fundamental of computer science and mathematics</b>  |
| <b>Learning Objective:</b> | <ul style="list-style-type: none"> <li>• To introduce concepts in automata theory and theory of computation.</li> <li>• To identify different formal language classes and their relationships.</li> <li>• To design grammars and recognizers for different formal languages.</li> </ul> |
| <b>Learning Outcome:</b>   | <ul style="list-style-type: none"> <li>• This course enables us to understand the concepts of theory of Computation and its applications.</li> </ul>  |

|  |
|--|
| <b>Paper-CSC204</b>  |
| <b>Theory of Computation</b>   |
| <b>UNIT-I:</b>   |
| Introduction: Automata theory, Computability theory, Complexity theory, Mathematical notations & terminology, Alphabet, String, Languages & operations on strings; Finite Automata (Deterministic): Formal definition, Transition function, Extended transition function, Language of DFA, Design of DFA; Finite Automata (Non-deterministic): Formal definition, Language of NFA, Equivalence of DFA & NFA; NFA with Epsilon Transition: Eliminating $\epsilon$ -transitions from NFA, Conversion from Epsilon-NFA to DFA, Minimization of DFA. |
| <b>UNIT-II:</b>  |
| Moore machines, Mealy machines; Regular expressions: Regular operators and their precedence, Building regular expressions, DFA to Regular expressions, Regular expressions to DFA, Arden's theorem, Pumping Lemma for Regular languages, Closure properties of Regular languages.  |
| <b>UNIT-III:</b>   |
| Introduction to Grammars: Definition, Derivation of string, Left and right linear grammars, Regular grammars; Context Free Grammars (CFG): Definition, Derivation of string, Language of CFG, Parse Tree, Ambiguity in grammar, Elimination of ambiguity, Normal forms of CFG: Chomsky and Greibach normal forms, Converting CFG to CNF & GNF, Closure properties of context free languages (CFL).   |
| <b>UNIT-IV:</b>  |
| Push Down Automata (PDA): PDA Components, Moves of a PDA, Design of a PDA, PDA to CFG and CFG to PDA conversion, Pumping lemma for CFL; Turing Machines (TM): Design of a TM, Variation of TM, Recursively Enumerable Languages and undecidable problems. Church Turing hypothesis, Recursive and recursively enumerable sets, Chomsky's hierarchy of languages. Ackermann's function, Godel numbering; NP Completeness: P and NP, NP complete and NP Hard problems.   |
| <b>Text Books:</b>   |
| <ol style="list-style-type: none"> <li>1. J. E. Hopcroft, R. Motwani, and J. D. Ullman, <b>Introduction to Automata Theory, Languages and Computation</b>, 3rd Edition, Pearson Education, 2007.</li> <li>2. P. Linz, <b>An Introduction to Formal Languages and Automata</b>, 4th Edition, Jones &amp; Bartlett Learning, 2006.</li> </ol>  |
| <b>Reference Books:</b>  |

1. M. Sipser, **Introduction to the Theory of Computation**, 3rd Edition, Cengage Learning, 2012.
2. J. C. Martin, **Introduction to Languages and the Theory of Computation**, 4th Edition, Tata McGraw-Hill, 2010.
3. K. L. P. Mishra, and N. Chandrasekaran, **Theory of Computer Science: Automata, Languages and Computation**, 3rd Edition, PHI, 2012.

## Paper-CSC205

### Lab on Java and OS

#### EXPERIMENTS ON JAVA:

1. 1. Programs to illustrate class and objects
2. Programs to illustrate Overloading & Overriding methods in Java.
3. Programs to illustrate constructors
4. Programs Illustrate the Implementation of Various forms of Inheritance. (Ex. Single, Hierarchical, Multilevel inheritance....)
- 5.. Program which illustrates the implementation of multiple Inheritance using interfaces in Java.
6. Program to illustrate the implementation of abstract class.
7. Programs to illustrate Exception handling
8. Programs to create Packages in Java.
9. Program to Create Multiple Threads in Java.
10. Program to Implement Producer/Consumer problem using synchronization
11. Developing a simple paint like program using applet
12. Developing programs on JButtons, JTextBox, JTextButton etc.

#### EXPERIMENTS ON OS:

1. Basics of UNIX commands.
2. Shell programming
3. Implementation of CPU scheduling. a) Round Robin b) SJF c) FCFS d) Priority
4. Implement all file allocation strategies :a) Sequential b) Indexed c) Linked
5. Implement Bankers algorithm for Dead Lock Avoidance
6. Implement an Algorithm for Dead Lock Detection
7. Implement the all page replacement algorithms a) FIFO b) LRU c) Optimal Page Replacement.
8. Implement Paging Technique of memory management

|                            |   |
|----------------------------|---|
| <b>Course Name:</b>        | <b>DATA WAREHOUSING AND MINING</b>  |
| <b>Category:</b>           | Programme Elective Course   |
| <b>Prerequisite:</b>       | <b>Data Structure and Algorithm, Linear Algebra, Basics of Web programming</b>  |
| <b>Learning Objective:</b> | This course deals with evolving multidimensional intelligent model from a typical system, representation of multi dimensional data for a data warehouse, discovering the knowledge imbibed in the high dimensional system, finding the hidden interesting patterns in data, and gives the idea to evaluate various mining techniques on complex data objects. |
| <b>Learning Outcome:</b>   | This course enables us to understand the concepts of Data Mining and its applications.  |

|  |
|--|
| <b>DSE Paper – CSC206A</b>   |
| <b>Data Warehousing and Mining</b>   |
| <b>UNIT-I:</b>   |
| Evolution of Decision Support Systems- Data warehousing Components –Building a Data warehouse, Data Warehouse and DBMS, Data marts, Metadata, Multidimensional data model, OLAP vs OLTP, OLAP operations, Data cubes, Schemas for Multidimensional Database: Stars, Snowflkes and Fact constellations.   |
| <b>UNIT-II:</b>  |
| Types of OLAP servers, 3–Tier data warehouse architecture, distributed and virtual data warehouses. Data warehouse implementation, tuning and testing of data warehouse. Data Staging (ETL) Design and Development, data warehouse visualization, Data Warehouse Deployment, Maintenance, Growth, Business Intelligence Overview- Data Warehousing and Business Intelligence Trends - Business Applications.   |
| <b>UNIT-III:</b>   |
| Data mining-KDD versus datamining, Stages of the Data Mining Process-task primitives, Data Mining Techniques -Data mining knowledge representation – Data mining query languages, Integration of a Data Mining System with a Data Warehouse – Issues, Data preprocessing – Data cleaning, Data transformation, Feature selection, Dimensionality reduction, Discretization and generating concept hierarchies-Mining frequent patterns- association-correlation.<br>Decision Tree Induction - Bayesian Classification – Rule Based Classification –Classification by Back propagation – Support Vector Machines – Associative Classification – Lazy Learners – Other Classification Methods. |
| <b>UNIT-IV:</b>  |
| Clustering techniques – , Partitioning methods- k-means Hierarchical Methods - distance-based agglomerative and divisible clustering,<br>Mining complex data objects, Spatial databases, temporal databases, Multimedia databases, Time series and Sequence data; Text Mining –Graph mining-web mining-Application and trends in data mining   |
| <b>Text Books:</b>   |
| 1. Jiawei Han and MichelineKamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, third edition2011, ISBN: 1558604898.<br>2. Alex Berson and Stephen J. Smith, “ Data Warehousing, Data Mining & OLAP”, TataMc Graw Hill Edition, Tenth Reprint 2007.<br>3.G. K. Gupta, “Introduction to Data Min Data Mining with Case Studies”, Easter EconomyEdition, Prentice Hall of India, 2006   |
| <b>Reference Books:</b>  |

1. Mehmedkantardzic, "Data mining concepts, models, methods, and algorithms", Wiley Interscience, 2003.
2. Ian Witten, Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques, third edition, Morgan Kaufmann, 2011.
3. George M Marakas, Modern Data Warehousing, Mining and Visualization, Prentice Hall, 2003

**Course Name:**

## **WIRELESS SENSOR NETWORKS**

**Category:**

Programme Elective Course

**Prerequisite:**

**Basic Computer Network**

**Learning Objective:**

The purpose of this course is to introduce students to Obtain a broad understanding about the network architecture of wireless sensor network. Understand all basic characteristics of wireless sensor networks and sensor nodes. The principles of data transmission, clustering algorithm and routing protocols. Design and development of new network architecture and MAC protocols.

**Learning Outcome:**

This course enables us to understand the concepts of wireless sensor network and its applications.

|   |
|---|
| <b>DSE Paper – CSC206B</b>  |
| <b>Wireless Sensor Networks</b>   |
| <b>UNIT-I:</b>  |
| <b>Introduction:</b> Networked wireless sensor devices, Applications: Habitat Monitoring, Smart Transportation, Key design challenges. <b>Network deployment:</b> Structured versus randomized deployment, Network topology, Connectivity. Introduction to cloud system, Sensor Cloud Systems, Challenges in Sensor Cloud Systems.  |
| <b>UNIT-II:</b>   |
| <b>Localization:</b> issues & approaches, Coarse-grained & Fine-grained node localization, Network-wide localization. <b>Wireless characteristics:</b> Basics, Wireless link quality, Radio energy considerations, SINR capture model for interference.   |
| <b>UNIT-III:</b>  |
| Issues in designing MAC protocol for WSNs, Classification of MAC Protocols, Energy efficiency in MAC protocols, Asynchronous sleep techniques, Sleep-scheduled techniques. Classification of Energy Management Schemes <b>Sleep-based topology control:</b> Constructing topologies for connectivity, constructing topologies for coverage.   |
| <b>UNIT-IV:</b>   |
| <b>Routing:</b> Metric-based approaches, Routing with diversity, Multi-path routing, Lifetime-maximizing energy-aware routing techniques, Geographic routing. Data-centric routing, Data-gathering with compression, Querying, Data-centric storage and retrieval, The database perspective on sensor networks.   |
| <b>Text Books:</b>  |
| 1. Wireless Sensor Networks: Technology, Protocols, and Applications: KazemSohraby, Daniel Minoli, TaiebZnati , Wiley Inter Science.<br>2. Networking Wireless Sensors: BhaskarKrismachari, Cambridge University Press  |
| <b>Reference Books:</b>   |
| 1. Wireless Sensor Networks: Architectures and Protocols: Edgar H. Callaway, Jr. Auerbach Publications, CRC Press.<br>2. Wireless Sensor Networks: Edited by C.S Raghavendra, Krishna M, Sivalingam, TaiebZnati , Springer.<br>3. Distributed Sensor Networks: A Multiagent Perspective, Victor Lesser, Charles L. Ortiz, and MilindTambe , Kluwer Publications.<br>4. Wireless Sensor Networks: An Information Processing Approach- by Feng Zhao, Leonidas Guibas , Morgan Kaufmann Series in Networking 2004. |

**Course Name:** **INTERNET OF THINGS**

**Category:** Program Elective Course

**Prerequisite:** Knowledge of Computer Networks

**Learning Objective:**

- To learn the concepts behind IoT and different application areas where sensors can be effectively used to capture real-time data for monitoring and control functions.
- To understand various protocols that govern the functioning of an IoT System

**Learning Outcome:**

- . Understand general concepts of Internet of Things (IoT)
- Recognize various devices, sensors and applications

|   |
|---|
| <b>DSE Paper –CSC 206C</b>  |
| <b>Internet of Things</b>   |
| <b>UNIT-I:</b>  |
| Introduction to IoT, Basic requirements for building an IoT system, IoT reference framework, IoT network level – performance criteria.<br>IoT devices: Sensors, Types of sensors and their functions: temperature, pressure, air pollution, proximity, infrared, moisture & humidity, flow, level, noise, and speed sensors. Characteristics of sensors. Use of RFID<br>Actuators, Types of actuators and their functions: electrical, mechanical, and hydraulic actuators, controlling IoT devices   |
| <b>UNIT-II:</b>   |
| IoT requirements for networking protocols, device addressing, credential management, wireless spectrum, determinism, security and privacy, application interoperability, semantic interoperability. IoT Protocol Stack: layered view.<br>Link layer: IEEE 802.15.4 technology, LoRaWAN end-to-end architecture, Time-Sensitive Networking<br>Internet Layer: Routing Protocol for Low-Power and Lossy Networks.   |
| <b>UNIT-III:</b>  |
| Application Protocols Layer: Data Serialization Formats, Communication Paradigms: Request/Response Versus Publish/Subscribe, Blocking Versus Non-blocking, QoS: Resource Utilization, Data Timeliness, Data Availability, Data Delivery<br>IoT Application Protocols: CoAP, XMPP, MQTT, AMQP, SIP, IEEE 1888, and DDS RTPS.<br>Application Services Layer: ETSI M2M network architecture, oneM2M standards.<br>IoT Services Platform: Functions and Requirements, IoT Platform Manager, Discovery, Communication Manager, Data Management, Management of IoT Devices, Configuration and Fault Management, Performance Management and measures |
| <b>UNIT-IV:</b>   |
| IoT security and Privacy: challenges, requirements, IoT Three-Domain Architecture, Attacks and Countermeasures for each domain.<br>Applications of IoT in areas like Smart home, Agriculture, Healthcare, Industry, Transportation, Retail, Oil and Gas, Energy etc. IoT Service Model: Anything as a Service, IoT Connected Ecosystems Models  |
| <b>Text Books:</b>  |
| 1. Internet of Things from Hype To Reality: The Road to Digitization (2nd ed), AmmarRayes and Samer Salam, Springer, 2019.  |



**Course Name:** **MICROPROCESSOR AND MICROCONTROLLER**

**Category:** Program Elective Course

**Learning Objective:** Timing diagram analyze role of microprocessor and microcontroller in computer systems,

Distinguish between maskable and non-maskable interrupt, and role of DMA in microprocessor.

**Learning Outcome:** Analyze working of 8086 and its architecture.

Learning the instructions used in 8086 & its application.

|   |
|---|
| <b>DSE Paper-CSC206D</b>  |
| <b>Microprocessor and Microcontroller</b>   |
| <b>UNIT-I:</b>  |
| <b>8085 AND 8086 MICROPROCESSORS</b><br>8085 Microprocessor: Architecture, Pin diagram, Physical memory organization, Timing diagrams, Interrupts of 8085, Instruction set and Assembly Language Programming of 8085.<br><br>8086 Microprocessor: Architecture, signal descriptions, common function signals, Minimum and Maximum mode signals, addressing modes, interrupt structure.                                |
| <b>UNIT-II:</b>   |
| <b>I/O INTERFACING</b><br>Interfacing with 8086/ 8085: Interfacing with RAMs, ROMs along with the explanation of timing diagrams. Interfacing with peripheral ICs like 8255, 8254, 8279, 8259, 8259 etc. Interfacing with key boards, LEDs, LCDs, ADCs, and DACs etc.<br><br>Introduction to microprocessors like 80386, 80486  |
| <b>UNIT-III:</b>  |
| <b>8051 MICROCONTROLLERS:</b> Overview of 8051 microcontroller Architecture. I/O Ports. Memory organization, addressing modes and instruction set of 8051, Interrupts, timer/Counter and serial communication, programming Timer Interrupts, programming external hardware interrupts, programming of serial communication interrupts, programming 8051 timers and counters, Introduction to other micro controllers. |
| <b>UNIT-IV:</b>   |
| <b>REAL WORLD INTERFACE DESIGN WITH 8051</b><br>Real world interface design: LED, SWITCH, keyboard, LCD, ADC, DAC, UART, RTC, PWM, Watch Dog Timer, DC Motor, Stepper Motors.   |
| <b>Text Books:</b>  |
| <ol style="list-style-type: none"><li>1. Advance Microprocessor and Peripherals, By, A.K. Roy and K.M. Bhurchandi, Tata McGraw-Hill Education</li><li>2. Mazidi and Mazidi, The 8051 Micro controller and Embedded Systems, pearson Education.</li><li>3. Microprocessor Architecture, Programming and Applications with the 8085, Ramesh Gaonkar, Penram International Publishing</li></ol>                          |

**Reference Books:**

1. The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware, and Applications , by Walter A. Triebel&Avtar Singh,
2. Micro processors and Interfacing, D. V. Hall, TMGH.  
The 8051 microcontroller , Kenneth. J. Ayala. Cengage learning.
3. Microcontrollers and application, Ajay. V. Deshmukh, TMGH

## Semester-III

**Course Name:** **COMPILER CONSTRUCTION**

**Category:** Programme Core Course

**Prerequisite:** **Theory of Computation / Automata theory**

**Learning Objective:**

1. To learn various stages of compilation, design phases of a compiler construction process.
2. This course will also introduce open source tool Lex and Yacc.

**Learning Outcome:**

1. Specify and analyse the lexical, syntactic and semantic structures of advanced language features
2. Separate the lexical, syntactic and semantic analysis into meaningful phases for a compiler to undertake language translation
3. Write a scanner, parser, and semantic analyser without the aid of automatic generators

|   |
|---|
| <b>Paper-CSC301</b>   |
| <b>Compiler Construction</b>  |
| <b>UNIT-I:</b>  |
| Compiler: Introduction – Programs related to compilers. Analysis of source program, Phases of compiler, modules related to compiler, grouping of phases. Lexical analysis – The role of Lexical Analyser. Input Buffering. Specification of Tokens. Recognition of Tokens. The Lexical-analyser Generator Lex.  |
| <b>UNIT-II:</b>   |
| Syntax Analysis – Introduction. Top-Down parsing, Brute Forcing, Recursive Descent, Predicative LL(1), Bottom-Up parsing : Shift reduce parsing, Introduction to LR Parsing, Powerful LR parsers: SLR, CLR, LALR, Parser Generators – Yacc. Error Recovery : Introduction, Error detecting and Reporting in various Phases.   |
| <b>UNIT-III:</b>  |
| Syntax Directed Translation – Syntax Directed Definitions. Evaluation Orders for SDDs. Applications of Syntax Directed Translation. Symbol Table Organization - Structure of Symbol table, Symbol Table organization, Data Structures of symbol Table.<br>Intermediate code generation: Variants of syntax trees. Three-Address Code, Types and Declarations. Translation of Expressions. Type Checking. Control Flow. Activation record, activation tree and run time storage management |
| <b>UNIT-IV:</b>   |
| Code Generation – Issues in the Design of a Code Generator. The Target Language. Addresses in the Target Code Basic Blocks and Flow Graphs. Optimization of Basic Blocks. Peephole Optimization. Register Allocation and Assignment. Machine Independent Optimizations – The Principal Sources of Optimizations, Introduction to data flow analysis, Foundation of data flow analysis.  |
| <b>Text Book:</b>   |
| 1. A. V. Aho, M. S. Lam, R. S. and J. D. Ullman, <b>Compilers: Principles, Techniques &amp; Tools</b> , 2 <sup>nd</sup> Edition, Pearson Education, 2007.   |
| <b>Reference Book:</b>  |
| 1. K. D. Cooper and L. Tarezon T. Munakata, <b>Engineering a Compiler</b> , 2 <sup>nd</sup> Edition, Elsevier, 2011.  |

**Course Name:**

# DATABASE MANAGEMENT SYSTEMS

**Category:**

Programme Core Course

**Prerequisite:**

**Basic Knowledge of Computer Programming and data structures**

**Learning Objective:**

- Classify modern and futuristic database applications based on size and complexity; design a database from understanding an Universe of Discourse, using ER diagrams; map ER model into Relational model and to normalize the relations; create a physical database from a design using DDL statements with appropriate key, domain and referential integrity constraints; analyze different ways of writing a query and justify which is the effective and efficient way; and compare and contrast various indexing strategies in different database systems and list key challenges in advanced database systems and to critique how they differ from traditional database systems.

**Learning Outcome:**

- To study the concepts of databases especially Relational Database design and query languages.

## Paper-CSC302

### Data Base Management Systems

#### UNIT-I:

**Introduction to DBMS:** Characteristics, Purpose, Application of the Database approach, Advantages of using DBMS approach upon file structure, Three-schema Architecture, Data Abstraction, Data Independence, Data base languages, DBMS Architecture, Data Models overview, Introduction to ER model and Relational data model.

**Relational Query Language:** Relational algebra, Tuple and Domain Relational Calculus, SQL.

#### UNIT-II:

**Database Design and ER model:** Overview of Design Process, Entities, Attributes, Constraints, Weak Entities, ER diagram, Extended ER Features, Reduction to Relational Schemas.

**Relational Database Design:** Feature of Good Relational Design, Atomic Domain and First Normal Form, Functional Dependency Theory, Decomposition of Schemas, Properties of Relational Decompositions, Normal forms and Normalization, 2NF, 3NF, BCNF, Multivalued Dependencies & 4NF. Performance tuning and Denormalization

#### UNIT-III:

**Query Processing and Optimization:** Evaluation of Relational Algebra Expression, QueryEquivalence, Join strategy, Query optimization algorithms. **Storage Strategies:** Indices, B+Trees, Hashing

#### UNIT-IV:

**Transaction Processing:** Transaction Concept, ACID Properties of Transaction, Serializability, Recoverability.

**Concurrency Control:** Overview, Lock-based Protocol, Timestamp ordering protocol, Multi version and Optimistic concurrency control techniques.

**Recovery Systems:** Database Failure and Recovery, Log based Recovery to preserve Atomicity and Durability

|  |
|--|
| <b>Text Book:</b>  |
| 1. A. Silberschatz, F. H. Korth, <b>Database System Concepts</b> , 6th Edition, MGH, 2010.   |
| <b>Reference Books:</b>  |
| 1. R. Elmasri, Fundamental of Database Systems, Pearson Education, 2008.<br>2. B. Desai, An Introduction to Database System, Galgotia publication.<br>3. C.J. Date, An Introduction to Database Systems, Pearson Education |

**Course Name:** **DESIGN AND ANALYSIS OF ALGORITHMS**

**Category:** Programme Core Course

**Prerequisite:** **Data Structure**

**Learning Objective:**

- Learn the algorithm analysis techniques.
- Become familiar with the different algorithm design techniques.
- Understand the limitations of Algorithm power.

**Learning Outcome:**

- Design algorithms for various computing problems and analyze the time and space complexity of algorithms.
- Choose appropriate data structures and algorithms, understand the ADT/libraries, and use it to design algorithms for a specific problem.
- Understand the necessary mathematical abstraction to solve problems and come up with analysis of efficiency and proofs of correctness.
- Comprehend and select algorithm design approaches in a problem specific manner and also modify existing algorithms to improve efficiency.

|  |
|--|
| <b>Paper-CSC303</b>  |
| <b>Design and Analysis of Algorithms</b>   |
| <b>UNIT-I:</b>   |
| Introduction to Design and analysis of algorithms, Growth of Functions (Asymptotic notations), Recurrences, Solution of Recurrences by substitution, Recursion tree method, Master Method, Analysis of Searching and Sorting Techniques: Brute Force Technique, Selection sort, Bubble sort.                     |
| <b>UNIT-II:</b>  |
| Divide and Conquer: Merge sort, Quick sort, Time complexity analysis for Merge and Quick sort. Transform and Conquer: Balanced search tree, Heaps and Heap sort. Dynamic Programming algorithms: Matrix Chain Multiplication, Elements of Dynamic Programming, Longest Common Subsequence, 0/1 Knapsack problem. |
| <b>UNIT-III:</b>   |
| Greedy Algorithms: Activity Selection Problem, Elements of Greedy Strategy, Fractional Knapsack Problem, Huffman Codes. Graph Algorithm - BFS and DFS, Minimum Spanning Trees, Kruskal algorithm, Prim's Algorithm, Single Source Shortest paths, Bellmen Ford Algorithm, Dijkstra's Algorithm.                  |
| <b>UNIT-IV:</b>  |
| String matching, Rabin-Karp Algorithm, KMP Algorithms. Theory of NP-completeness: Complexity classes of P, NP, NP-Hard, NP complete. Polynomial reduction, Cook's theorem, discussion on SAT, CNF-SAT, Min vertex cover, max clique, Graph coloring.   |
| <b>Text Book:</b>  |
| 1. T.H.Coreman et.al. <b>Introduction to Algorithms</b> ,MIT press Cambridge, 2001.  |
| <b>Reference Books:</b>  |
| 1.M. R. Kabat, <b>Design and Analysis of Algorithms</b> , PHI, 2013.<br>2. S. Sridhar, <b>Design and Analysis of Algorithms</b> , Oxford University Press<br>3. E. Horowitz, S. Sahni, <b>Fundamentals of Computer Algorithms</b> , computer science press, 1978.  |

**Course Name:** WEB TECHNOLOGY

**Category:** Programme Core Course

**Prerequisite:** Good understanding of Object Oriented Programming

- Basic programming skills

**Learning Objective:**

- Understand the principles of creating an effective web page, including an in-depth consideration of information architecture.
- Become familiar with graphic design principles that relate to web design and learn how to implement theories into practice.
- Learn the language of styling the web page using HTML and CSS.
- Acquire knowledge to develop valid and well-formed XML document.
- Learn techniques of responsive web application design.
- Develop basic programming skills using JavaScript and PHP.

**Learning Outcome:**

- Design and implement dynamic websites with good aesthetic sense of designing.
- Have a Good grounding of web application terminologies and web development tools.
- Develop multiplatform interactive and dynamic web applications.
- Outline the key components that facilitate the interoperability nature of web services.

|  |
|--|
| <b>Paper-CSC304</b>  |
| <b>Web Technology</b>  |
| <b>UNIT-I:</b>   |
| <b>Web Essentials: Clients, Servers, and Communication.</b> The Internet-Basic Internet Protocols - The World Wide Web-HTTP request message-response message-Web Clients Web Servers-Case Study. <b>Markup Languages: XHTML.</b> An Introduction to HTML History-Versions-Basic XHTML Syntax and Semantics-Some Fundamental HTML Elements-Relative URLs-Lists-tables-Frames-Forms-XML Creating HTML Documents-Case Study.  |
| <b>UNIT-II:</b>  |
| <b>Style Sheets:</b> CSS-Introduction to Cascading Style Sheets-Features-Core Syntax-StyleSheets and HTML Style Rule Cascading and Inheritance-Text Properties-Box Model-Normal Flow Box Layout-Beyond the Normal Flow-Other Properties-Case Study. <b>Client-Side Programming:</b> The JavaScript Language-History and Versions Introduction to JavaScript in Perspective-Syntax-Variables and Data Types-Statements-Operators-Literals-Functions-Objects-Arrays-Built-in Objects-JavaScript Debuggers. |
| <b>UNIT-III:</b>   |
| <b>PHP:</b> Introducing PHP, PHP Language Basics–Using variables, Understanding Data Types, Operators and Expressions, Constants. Decisions and Loops–Making Decisions, Doing Repetitive Tasks with Looping, Mixing Decisions and Looping with HTML. <b>Strings</b> –Creating and Accessing Strings, Searching Strings, Replacing Text with strings, Dealing with Upper and Lowercase, Formatting Strings. <b>Arrays</b> –Creating Arrays, Accessing Array Elements, Looping                             |

Through Arrays with for-each, Working with Multidimensional Arrays, Manipulating Arrays. **Functions**, writing your own Functions, Working with References, Writing Recursive Functions.

Objects–Introduction OOP Concepts, Creating Classes and Objects in PHP, Creating and using Properties, Working with Methods.

#### **UNIT-IV:**

**PHP MySQL:** Handling HTML Forms with PHP–How HTML form works, Capturing Form Data with PHP, Dealing with Multi-Value Fields, Generating Web Forms with PHP, Storing PHP Variables in Forms, Creating File Upload Forms, Redirecting After a Form Submission. **Introducing Databases and SQL**–Deciding How to Store Data, Understanding Relational Databases, Setting Up MySQL, A Quick Play with MySQL, Connecting MySQL from PHP. **Retrieving Data from MySQL with PHP**–Setting Up the Book Club Database, Retrieving Data with SELECT, Creating a Member Record Viewer. Manipulating MySQL Data with PHP–Inserting, Updating, and Deleting Records.

#### **Text Books:**

1. M. Doyle, **Beginning PHP 5.3**, 1<sup>st</sup> Edition, John Wiley & Sons, 2011.
2. J. Duckett, **Beginning HTML, XHTML, CSS and JavaScript**, 1<sup>st</sup> Edition, John Wiley & Sons, 2011

#### **Reference Books:**

1. L. Welling, L. Thomson, **PHP and MySQL Web Development**, 1<sup>st</sup> Edition, Sams Publishing, 2003.



**Lab on Web Technology and DBMS**

**EXPERIMENTS ON WEB TECHNOLOGY:**

1. Design a web page for your college containing a description of courses, departments, faculties using different HTML elements.
2. Customize the HTML page using CSS.
3. Create a login form which will check for username and password. If login successful then goto next form(Student registration form) that contains form with fields Name, Email, Mobile No ,Gender and a button .write a JavaScript code to validate data of above form.
4. Develop simple calculator for addition, subtraction, and multiplicationanddivisionoperation using JavaScript.
5. Design a web page to create your resume usecolor, textcolor, an Image, font etc. You may use CSS to format web page.
6. Create user Student registrationform (use textbox, checkbox, radiobutton, select box etc.)
7. Design an examination registration form using HTML. Sore the required data in a database (create it using MySQL) using PHP and also display message regarding status of registration (Success or Unsuccess).
8. Create a database through PHP and MySQL, and create, delete and modify data on database.
9. Store the data from a HTML form designed for registering a webinar and using PHP and MySQL, store, and update the data. Display the database data in HTML form.
10. Create anapplication using HTML, PHP. Create login form using HTML and checkusername and password using PHP, if login successful it will go on next HTML page and if failure again goesback to login page.

**EXPERIMENTS ON DBMS:**

1. Creation of a tables using create command and writing SQL queries to retrieve information from the tables.
2. Implement data definition languages (Create, Alter, Drop, Truncate, and Rename) &data manipulation languages (Insert, Update, and Delete) for updating and viewing records.
3. Implement SELECT command with different clauses (where clause, having clause, group by clause, order by clause).
4. Implement Single Row function (character, numeric, data functions).
5. To implement Group function (AVG, MIN, MAX, SUM).
6. Implement various types of integrity constraints (NOT NULL Constraint, DEFAULT Constraint, UNIQUE Constraint, PRIMARY Key, FOREIGN Key, CHECK Constraint).
7. Creation of Views, Synonyms, Sequence, Indexes, Save point.
8. Creating relationship between tables.
9. Implementation of PL/SQL block.
10. Write a PL/SQL block to satisfy some conditions by accepting input from theuser.
11. Write a PL/SQL block that handles all types of exceptions.

Course Name: **NETWORK AND INTERNET TECHNOLOGIES**

Category: **IDSE Course**

**Prerequisite:** Basic knowledge of a computer system and Internet is required.

**Learning Objective:**

- study the fundamental concepts of computer networks.
- Introduce the fundamental concepts of Web Design.
- expose students to develop basic web applications.

**Learning Outcome:**

- describe the basics of computer networks topology.
- understand the basic concept of transmission media, LAN topology and network devices,
- develop web applications using web technologies

|   |
|---|
| <b>IDSE-Paper-CSC306A</b>   |
| <b>Network and Internet Technologies</b>  |
| <b>UNIT-I:</b>  |
| <b>Computer Networks:</b> Introduction to computer network, datacommunication, components of data communication, data transmission mode, data communication measurement, LAN, MAN, WAN, wireless LAN, internet, intranet,extranet. <b>Network Models:</b> Client/ server network and Peer-to-peer network, OSI, TCP/IP, layersandfunctionalities.   |
| <b>UNIT-II:</b>   |
| <b>Transmission Media:</b> Introduction, Guided Media: Twisted pair, Coaxial cable, Opticalfiber. Unguided media: Microwave, Radio frequency propagation,Satellite. <b>LAN Topologies:</b> Ring, bus, star, mesh and treetopologies. <b>Network Devices:</b> NIC, repeaters, hub, bridge, switch, gateway androuter. <b>Internet Terms:</b> Web page, Home page, website, internet browsers, URL, Hypertext, ISP, Web server, download and upload, online andoffline. |
| <b>UNIT-III:</b>  |
| <b>Introduction to Web Design:</b> Introduction to hypertext markup language (html) Document type definition, creating web pages, lists, hyperlinks, tables, web forms, inserting images, frames, hosting options and domain name registration.   |
| <b>UNIT-IV:</b>   |
| <b>Style Sheets:</b> CSS-Introduction to Cascading Style Sheets-Features-Core Syntax-StyleSheets and HTML Style Rule Cascading and Inheritance-Text Properties-Box Model-Normal Flow Box Layout-Beyond the Normal Flow-Other Properties-Case Study.<br><b>Client-Side Programming:</b> The JavaScript Language-History and Versions Introduction to   |

JavaScript in Perspective-Syntax-Variables and Data Types-Statements-Operators-Literals-Functions-Objects-Arrays-Built-in Objects-JavaScript Debuggers.

**Text Book:**

1. J. A. Ramalho, **Learn Advanced HTML 4.0 with DHTML**, BPB Publications,2007

**Reference Books:**

1. B. A. Forouzan, Data Communication and Networking , Tata McGrawHill, 2008.
2. D.R. Brooks, An Introduction to HTML and Javascript for Scientists and Engineers, Springer, 2007.
3. HTML A Beginner's Guide, Tata McGraw-Hill Education,2009.
4. J. A. Ramalho, **Learn Advanced HTML 4.0 with DHTML**, BPB Publications,2007

Course Name **COMPUTER FUNDAMENTALS**

Category IDSE Course

Prerequisite: Basic mathematics

Learning Objective:

- To make students understand and learn the basics of computer.
- To make them familiar with the parts and functions of computer.

Learning Outcome:

- . Identify and describe the functionality of various parts of digital computer. Describe the working principle of Computer. Perform various binary arithmetic operations. Describe the use of different type of memory used in computer

|  |
|--|
| <b>IDSE-Paper-CSC306B</b>  |
| <b>Computer Fundamentals</b>   |
| .  |
| <b>UNIT-I:</b>   |
| <b>Computer Basics:</b> Simple model of computer, Problem solving using computer (flowchart, program, working of a computer, hardware and software). <b>Data Representation:</b> Character representation, representation of integers and fractions, Decimal to binary conversion. <b>Input / Output Units.</b>  |
| <b>UNIT-II:</b>  |
| <b>Memory System:</b> Basic Concepts RAM, ROM, Speed, size and cost, Cache Memory concepts, Cache Memory mapping techniques, Virtual Memory concepts, Secondary Storage. <b>Processor:</b> Structure of instructions, Description of a processor, Machine language program, Algorithm to simulate the hypothetical computer.   |
| <b>UNIT-III:</b>   |
| <b>Binary Arithmetic:</b> Addition, subtraction, signed numbers, Two's complement representation of numbers, addition/subtraction of numbers in 2's complement notation, binary multiplication, binary division, floating point representation of numbers, arithmetic operation with normalized floating point numbers.  |
| <b>UNIT-IV:</b>  |
| <b>Logic circuits:</b> switching circuits, AND, OR, NOT operation, Boolean functions, canonical forms for Boolean function, logic circuits. <b>Computer Architecture:</b> Interconnection of Units, Processor to memory communication, I/O devices to processor communication, Bus architecture of personal computers; <b>Introduction to Programming Language, Operating system..</b> |
| <b>Text Book:</b>  |
| 1. V. Rajaraman, and N. Adabala, <b>Fundamentals of computers</b> , PHI, 2014.<br>2. A. Goel, <b>Computer Fundamentals</b> , Pearson Education, 2010.  |
| <b>Reference Book:</b>   |
| 1. P. Aksoy, L. DeNardis, <b>Introduction to Information Technology</b> , Cengage Learning, 2006.<br>2. P. K. Sinha, P. Sinha, <b>Fundamentals of Computers</b> , BPB Publishers, 2007.  |

## Course Name: **INTRODUCTION TO PROGRAMMING USING PYTHON**

### Category: IDSE Course

**Prerequisite:** Basic analytical and logical understanding including basic knowledge and usage of computers is required for this course. Prior experience with any other programming language will be beneficial.

#### **Learning Objective:**

- Introduce Python programming to students.
- Apply problem solving techniques to solve computational problems using python.
- Expose students to develop application for solving computational problems.

#### **Learning Outcome::**

- Able to install and set the python environment in their PC and execute python programs.
- Proficiently use functions and core data structure like list, dictionaries, tuple.
- Understand Python syntax, flow control, and functions to solve real life application.
- Develop application using Object Oriented Programming concepts of Python.

|  |
|--|
| <b>IDSE-Paper-CSC306C</b>  |
| <b>Introduction to Programming Using Python</b>  |
| <b>UNIT-I:</b>   |
| <b>Introduction:</b> Installation, First Python Program: Interactive Mode Programming, Script Mode Programming; Identifiers, Reserved Words, Lines and Indentation, Multi-Line Statements, Quotation & Comments; Assigning Values to Variables, Multiple Assignment.   |
| <b>UNIT-II:</b>  |
| <b>Standard Data Types:</b> Numbers, Strings, Lists, Tuples, Dictionary; Data Type Conversion; Basic Operators: Arithmetic, Comparison, Assignment, Bitwise; Operators: Logical, Membership, Identity; Operators Precedence; Python Numbers & Mathematical functions<br><b>Data Type Conversion:</b> Basic Operators: Arithmetic, Comparison, Assignment, Bitwise; Basic Operators, Python Numbers & Mathematical functions; Python Strings. |
| <b>UNIT-III:</b>   |
| <b>Python statements and Loops:</b> if, if-else, While, for loops, break, continue, pass, Python Function; Files I/O.<br><b>Functions:</b> Definition, call, positional and keyword parameter. Default parameters, variable number of arguments. Modules - import mechanisms. Functional programming - map, filter, reduce, max, min. lambda function - list comprehension   |
| <b>UNIT-IV:</b>  |
| <b>Object Oriented Programming:</b> classes and objects - Inheritance – Polymorphism overloading; Error handling & Exceptions - try, except and raise - exception propagation<br><b>File Processing:</b> reading and writing files.  |
| <b>Text Book:</b>  |
| .1. Python Programming Fundamentals - A Beginner's Handbook, NischaykumarHegde   |
| <b>Refrence Book:</b>  |
| 1.Python: The Complete Reference by Martin C.Brown   |

## Semester-IV

Course Name: **MACHINE LEARNING**

Category: Programme Core Course

**Prerequisite: Basic knowledge of Mathematics**

**Learning Objective:**

- learn the fundamental concepts behind supervised, unsupervised & reinforcement learning,
- understand the concept behind dimensionality reduction in case of high dimensional data.
- assess & select appropriate model and use cross validation to tune their parameters.

**Learning Outcome:**

- Analyze the concepts of supervised unsupervised & reinforcement learning and its functionalities.
- Perform classification using Bayes classifier, Neural network, K-nn, SVM, and Decision Tree.
- Reduce dimensionality using feature selection and extraction.
- Determine most appropriate model in a specific context using model cross validation techniques

|   |
|---|
| <b>Paper-CSC401</b>   |
| <b>Machine Learning</b>   |
| <b>UNIT-I:</b>  |
| Introduction to AI and intelligent agents. Problem Solving : Solving Problems by Searching, heuristic search techniques.<br>Learning ((Supervised, Unsupervised and Reinforcement learning), Learning Models (Classification, Regression, Clustering).<br>Cluster Analysis, Partitioning Methods (k-Means, k-Medoids), Hierarchical Methods, Density-Based Methods, Evaluation of Clustering. |
| <b>UNIT-II:</b>   |
| Conditional Probability, Bayes' Theorem, Naïve Bayes Classifier, K-nearest neighbour, Multiple linear regression, Shrinkage method, Ridge regression, Logistic regression, Linear Discriminant Analysis.  |
| <b>UNIT-III:</b>  |
| Neural Networks - Introduction, Early Models, Perceptron Learning, Neural Networks - Backpropagation, Neural Networks - Initialization, Training & Validation, Parameter Estimation. Decision Tree, Decision Tree Induction, Attribute Selection Measures, Information Gain, Gain Ratio, ID3, C4.5, Gini Index, CART.   |

#### **UNIT-IV:**

Support Vector Machine for linearly separable data, Kernel function, Support Vector Machine for linearly non-separable data.

Dimensionality reduction, Feature selection, Feature extraction, Principal Component Analysis. Model Cross-validation, Performance of Classification algorithms (Confusion Matrix, Precision and Recall).

#### **Text Book:**

- T1. Trevor Hastie, Robert Tibshirani, Jerome Friedman, **The Elements of Statistical Learning-Data Mining, Inference, and Prediction**, Springer Verlag Second Edition, 2009.
- T2. Simon Haykin, **Neural Networks and Learning Machines**, Pearson Education, third edition, 2009.

#### **Reference Book:**

- R1. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, **An introduction to statistical learning with applications in R**, Springer, 2013.
- R2. C. M. Bishop, **Pattern Recognition and Machine Learning**, Springer, 2006.

**Course Name:** SOFTWARE ENGINEERING & OOAD

**Category:** Programme Core Course

**Prerequisite:** Knowledge on programming and data structure

- Learning Objective:**
- To understand common cycle process life processes.
  - To understand the basic concepts in Requirement engineering, software design, coding, testing and maintenance
  - To learn about the role of project management including scheduling, planning, risk management etc.
  - To have a basic knowledge about software quality, how to ensure good quality software.

- Learning Outcome:**
- After successful completion of the course the students will be able to demonstrate basic software engineering methods and practices, and their appropriate application.

|  |
|--|
| <b>Paper-CSC402</b>  |
| <b>Software Engineering &amp; OOAD</b>   |
| <b>UNIT-I:</b>   |
| <b>Software and software engineering:</b> Basic concepts about software and program, the nature of software, Evolution of Software Engineering, Stakeholders in software engineering, Software quality, Software engineering projects, Activities common to software projects, Basic concept on process and life cycle models.   |
| <b>UNIT-II:</b>  |
| <b>Models:</b> Waterfall, Prototype, Evolutionary, Incremental, Spiral, V-model, RAD.<br><b>Requirement Analysis:</b> System and software requirements, Types of software requirements, Functional and non-functional requirements, Domain requirements, User requirement Elicitation and analysis of requirements, Overview of requirement techniques, Viewpoints, Interviewing, Scenario, Requirement validation, Requirement specification, Software requirement Specification (SRS) Structure and contents, SRS format   |
| <b>UNIT-III:</b>   |
| <b>Introduction to Object Oriented Technology:</b> Development and OO Modeling History, Modeling Concepts,<br><b>Object Oriented Analysis:</b> Identifying Use-Cases, Complexity in Object Oriented Analysis, Business Process Modeling and Business Object Analysis, Use-Case Driven Object Oriented Analysis, Use-Case Model.<br><b>Class Modeling:</b> Object and class concepts, link and association, Generalization and Inheritance, Advanced class modeling- aggregation, Abstract class, constraints. State Modeling: Event, state, Transition and conditions, state diagram, state diagram behaviour, concurrency, Relation of Class and State models. Interaction Modeling: sequence models, activity diagrams |
| <b>UNIT-IV:</b>  |
| <b>Software Project Management:</b> Overview of Project Management, Responsibilities of Project Manager, Project Planning, Metrics for Project Size Estimation, Factors Influencing Project Management, Project Estimation Techniques, COCOMO Model and its versions, Scheduling, Work Breakdown Structures (WBS), Activity Network, Critical Path Method (CPM), Program Evaluation and Review Technique (PERT) Chart, GANTT Chart, Risk Management, Configuration Management  |



**Software Testing:** Testing overview, concepts, Scope of Testing, Testing Constraints, Testing Life Cycle, Levels of Testing, Blackbox Testing, Whitebox Testing, Integration Testing, Testing Object Oriented Programs

**Text Books:**

1. R.S. Pressman, "Software Engineering", A Practitioner's Approach, 7/e, McGraw-Hill, 2009
2. Timothy C. Lethbridge, Robert Laganière, "Object-Oriented Software Engineering Practical Software development using UML and Java", McGraw-Hill, Second Edition.

**Reference Books:**

1. Sommerville, "Software Engineering", 9/e, Addison Wesley.
2. R. Mall, "Fundamentals of Software Engineering", 3/e, PHI

|                     |                             |
|---------------------|-----------------------------|
| <b>Course Name:</b> | <b>INFORMATION SECURITY</b> |
| <b>Category:</b>    | Programme Core Course       |

**Prerequisite:** **Computer Network**

- Learning Objective:**
- Explain the objectives of information security
  - Explain the importance and application of each of confidentiality, integrity, authentication and availability
  - Understand various cryptographic algorithms.
  - Understand the basic categories of threats to computers and networks
  - Describe public-key cryptosystem.
  - Describe the enhancements made to IPv4 by IPSec
  - Understand Intrusions and intrusion detection
  - Discuss the fundamental ideas of public-key cryptography.
  - Discuss about Web security and Firewalls.

- Learning Outcome:**
- Student will be able to understand basic cryptographic algorithms, message and web authentication and security issues.
  - Ability to identify information system requirements for both of them such as client and server.
  - Ability to understand the current legal issues towards information security.

|  |
|--|
| <b>Paper-CSC403</b>  |
| <b>Information Security</b>  |
| <b>UNIT-I:</b>   |
| <p><b>Attacks on Computers and Computer Security:</b> Introduction, The need for security, Security goals, Security attacks(Attack on Confidentiality,Integrity,Availability)Security Services and Mechanisms, Techniques(Cryptography,Steganography).</p> <p>Introduction to plain text and cipher text, encryption and decryption. substitution techniques, transposition techniques, symmetric and asymmetric key cryptography, steganography, possible types of cryptanalysis attacks.</p>   |
| <b>UNIT-II:</b>  |
| <p><b>Symmetric key Ciphers:</b> Block Cipher principles &amp; Algorithms(DES, AES, Blowfish), Differential and Linear Cryptanalysis, Block cipher modes of operation, Stream ciphers RC4, Location and placement of encryption function.</p> <p><b>Introduction to number theory-</b>Prime numbers,Euler's Phi-Function,Fermat's and Euler's theorem, Chinese Remainder Theorem,Generating Primes(MersennePrime,Fermat Prime),Primality testing(Deterministic algorithms,Probabilistic algorithms)</p> <p><b>Asymmetric key Ciphers:</b> Principles of public key cryptosystems, Algorithms(RSA, Diffie-Hellman), Key Distribution.</p> |
| <b>UNIT-III:</b>   |
| <p><b>Message Authentication Algorithms and Hash Functions:</b>Message authentication (MDC,MAC)Nested MAC,HMAC,CMAC,Whirlpool. Hash functions: MD5 Message Digest algorithm,SHA-1. Digital signatures, Authentication Applications: Kerberos, X.509 Authentication Service, Public — Key Infrastructure, Biometric Authentication.</p>   |

#### **UNIT-IV:**

**E-Mail Security:** Pretty Good Privacy, S/MIME **IP Security:** IP Security overview, IP Security architecture, Authentication Header, Encapsulating security payload, Combining security associations, key management.

**Web Security:** Web security considerations, Secure Socket Layer and Transport Layer Security, Secure electronic transaction. Intrusion Detection System(types, techniques).

**Intruders, Virus and Firewalls:** Intruders, Intrusion detection, password management, Virus and related threats, Countermeasures, Firewall design principles, Types of firewalls.

#### **Text Book:**

1. B. A. Forouzan, D. Mukhopadhyay, **Cryptography and Network Security**, 2<sup>nd</sup> Edition, McGraw Hill, 2007.

#### **Reference Books:**

1. A. Kahate, **Network Security**, 2<sup>nd</sup> Edition, McGraw Hill, 2008.
2. W. Stallings, **Cryptography and Network Security**, 4<sup>th</sup> Edition, Pearson Education, 2006.

Course Name: **CLOUD COMPUTING**

Category: Programme Core Course

Prerequisite: **Computer Networks and Operating System**

**Learning Objective:**

- Provides a comprehensive study of cloud computing
- Know the different models of distributed system
- Know the design issues of cloud computing platforms
- Understands the workflow of service-oriented architectures
- Analyze the different cloud computing resource management policies

**Learning Outcome:**

- Ability to understand various service delivery models of a cloud computing architecture.
- Knows the design issues of cloud computing platforms
- Understand the concepts of service-oriented architecture
- Analyze the different workflows of service-oriented architecture
- Ability to understand the security challenges and address the challenges
- Understand the ways in which the cloud can be programmed and deployed

|  |
|--|
| <b>Paper-CSC404</b>  |
| <b>Cloud Computing</b>   |
| <b>UNIT-I:</b>   |
| <b>Distributed System Models and Enabling Technologies:</b> scalable computing over the Internet, technologies for network-based systems, system models for distributed and cloud computing, software environments for distributed systems and clouds, performance, security, and energy efficiency.                                     |
| <b>UNIT-II:</b>  |
| <b>Virtual Machines and Virtualization of Clusters and Data Centers:</b> implementation levels of virtualization, virtualization structures/tools and mechanisms, virtualization of CPU, memory and I/O devices, virtual clusters and resource management, virtualization of data-center automation.                                     |
| .  |
| <b>UNIT-III:</b>   |
| <b>Cloud Platform Architecture over Virtualized Data Centers:</b> cloud computing and service models, data-center design and interconnection networks, architecture design of compute and storage clouds, public cloud platforms: GAE, AWS (EC2 and S3) and Azure, inter-cloud resource management, cloud security and trust management. |

## UNIT-IV:

**Cloud Programming and Software Environments:** features of cloud and grid platforms, parallel and distributed programming paradigms, programming support of Google App Engine, Programming on Amazon AWS and Microsoft Azure, emerging cloud software environments

**Advanced Topics in Cloud Computing and Applications:** Energy efficiency in clouds, market-based management of clouds, federated clouds/intercloud, third-party cloud services, scientific applications: healthcare, biology, geoscience and business and consumer applications.

### Text Books:

1. Kai Hwang, Geoffery C. Fox and Jack J. Dongarra, “Distributed and Cloud Computing: Clusters, Grids, Clouds and the Future of Internet”, First Edition, Morgan Kaufman Publisher, an imprint of Elsevier, 2012.
2. Rajkumar Buyya, Christian Vecchiola and S. ThamaraiSelvi, “Mastering Cloud Computing: Foundations and Applications Programming”, MK Publisher, Elsevier, 2013

### Reference Books:

1. Tom White, “Hadoop The Definitive Guide”, First Edition. O’Reilly, 2009.
2. Ian Foster, Carl Kesselman, “The Grid: Blueprint for a New Computing Infrastructure”, 2nd Edition, Morgan Kaufmann.
3. P. K. Pattnaik, M. R. Kabat and S. Pal, Fundamentals of Cloud Computing, Vikas Publishing House Pvt. Ltd., 2015

## Paper-CSC405

### Project Work Report and VIVA VOCE

Web-development project: Students must follow software engineering principles to make the project.

Research project: Students can take a research work, review the related literatures, then propose a method or implement an existing one.