## Revised Course Structure Academic Year 2016-17

## Course Structure of M. Sc. [Computer Science]
## w.e.f. (Academic Year 2016-17)

| Semester-I | | | | Total Marks | |
|---|---|---|---|---|---|
| Course Code | Course Title | No. of Credits | No. of Hrs./Week | Internal | External |
| CSC401 | Constitution of India | 2 | 2 | 10 | 40 |
| CSC402 | Research Methodology | 2 | 2 | 10 | 40 |
| CSC403 | Advance Operating System | 3 | 3 | 20 | 80 |
| CSC404 | Data Structures and Algorithms | 3 | 3 | 20 | 80 |
| CSC405 | Discrete Mathematical Structures | 3 | 3 | 20 | 80 |
| CSC406 | Programming in Core Java | 3 | 3 | 20 | 80 |
| CSC451 | Practical based on CSC403 | 2 | 4 | - | 50 |
| CSC452 | Practical based on CSC404 | 2 | 4 | - | 50 |
| CSC453 | Practical based on CSC 405 | 2 | 4 | - | 50 |
| CSC454 | Practical based on CSC406 | 2 | 4 | - | 50 |
| Total No. of Credits | | 24 | - | - | - |

| Semester-II | | | | Total Marks | |
|---|---|---|---|---|---|
| Course Code | Course Title | No. of Credits | No. of Hrs./Week | Internal | External |
| CSC407 | Research Project Review Writing | 1 | 1 | 25 | - |
| CSC408 | Relational DBMS & SQL | 3 | 3 | 20 | 80 |
| CSC409 | Software Engineering and CASE Tools | 3 | 3 | 20 | 80 |
| CSC410 | Compiler Design | 3 | 3 | 20 | 80 |
| CSC411 | Advance Java | 3 | 3 | 20 | 80 |
| CSC412 | Computer System Architecture | 3 | 3 | 20 | 80 |
| CSC455 | Practical based on CSC408 | 2 | 4 | - | 50 |
| CSC456 | Practical based on CSC409 | 2 | 4 | - | 50 |
| CSC457 | Practical based on CSC 410 | 2 | 4 | - | 50 |
| CSC458 | Practical based on CSC411 | 2 | 4 | - | 50 |
| CSC459 | Practical based on CSC412 | 2 | 4 | - | 50 |
| Total No. of Credits | | 26 | - | - | - |

| Semester-III | | | | Total Marks | |
|---|---|---|---|---|---|
| Course Code | Course Title | No. of Credits | No. of Hrs./Week | Internal | External |
| CSC501 | Advance Computer Networks | 3 | 3 | 20 | 80 |
| CSC502 | Computer Graphics | 3 | 3 | 20 | 80 |
| - | Elective –I (Generic) | 3 | 3 | 20 | 80 |
| CSC551 | Project/Dissertation Part –I | 12 | 24 | 100 | 200 |
| CSC552 | Practical based on CSC501 | 2 | 4 | - | 50 |
| CSC553 | Practical based on CSC 502 | 2 | 4 | - | 50 |
| - | Practical based on Elective-I | 2 | 4 | - | 50 |
| Total No. of Credits | | 27 | - | - | - |

| Semester-IV | | | | Total Marks | |
|---|---|---|---|---|---|
| Course Code | Course Title | No. of Credits | No. of Hrs./Week | Internal | External |
| CSC503 | Pattern Recognition | 3 | 3 | 20 | 80 |
| - | Elective –II (Discipline Centric) | 3 | 3 | 20 | 80 |
| CSC554 | Project/Dissertation Part -II | 12 | 24 | 100 | 200 |
| CSC555 | Seminars | 1 | 2 | - | 25 |
| - | Service Course | 4 | 4 | 20 | 80 |
| CSC556 | Practical based on CSC503 | 2 | 4 | - | 50 |
| - | Practical based on Elective-II | 2 | 4 | - | 50 |
| Total No. of Credits | | 27 | - | - | - |

| Elective-I | | | | Total Marks | |
|---|---|---|---|---|---|
| Course Code | Course Title | No. of Credits | No. of Hrs./Week | Internal | External |
| CSC521 | Digital Signal Processing | 3 | 3 | 20 | 80 |
| CSC522 | Digital Image Processing | 3 | 3 | 20 | 80 |
| CSC523 | Information Theory | 3 | 3 | 20 | 80 |
| CSC524 | Soft Computing | 3 | 3 | 20 | 80 |
| CSC525 | Data Mining | 3 | 3 | 20 | 80 |
| CSC526 | Network Security | 3 | 3 | 20 | 80 |
| CSC527 | Mobile Computing | 3 | 3 | 20 | 80 |
| CSC557 | Practical Based on CSC521 | 2 | 4 | - | 50 |
| CSC558 | Practical Based on CSC522 | 2 | 4 | - | 50 |
| CSC559 | Practical Based on CSC523 | 2 | 4 | - | 50 |
| CSC560 | Practical Based on CSC524 | 2 | 4 | - | 50 |
| CSC561 | Practical Based on CSC525 | 2 | 4 | - | 50 |
| CSC562 | Practical Based on CSC526 | 2 | 4 | - | 50 |
| CSC563 | Practical Based on CSC527 | 2 | 4 | - | 50 |

| Elective-II | | | | Total Marks | |
|---|---|---|---|---|---|
| Course Code | Course Title | No. of Credits | No. of Hrs./Week | Internal | External |
| CSC528 | Data Warehousing | 3 | 3 | 20 | 80 |
| CSC529 | Biometrics and Security Systems | 3 | 3 | 20 | 80 |
| CSC530 | Cloud Computing | 3 | 3 | 20 | 80 |
| CSC531 | Decision Support System | 3 | 3 | 20 | 80 |
| CSC532 | Remote Sensing and GIS | 3 | 3 | 20 | 80 |
| CSC533 | Human Computer Interaction | 3 | 3 | 20 | 80 |
| CSC534 | Computer Vision | 3 | 3 | 20 | 80 |
| CSC564 | Practical Based on CSC528 | 2 | 4 | - | 50 |
| CSC565 | Practical Based on CSC529 | 2 | 4 | - | 50 |
| CSC566 | Practical Based on CSC530 | 2 | 4 | - | 50 |
| CSC567 | Practical Based on CSC531 | 2 | 4 | - | 50 |
| CSC568 | Practical Based on CSC532 | 2 | 4 | - | 50 |
| CSC569 | Practical Based on CSC533 | 2 | 4 | - | 50 |
| CSC570 | Practical Based on CSC534 | 2 | 4 | - | 50 |

**Total (I+II+III+IV) Semester (24+26+27+27) Credits = 104**

# Detailed Syllabus

## *Semester – I*

## 1. Research Methodology

| | | | |
|---|---|---|---|
| **Course Code** | **CSC402** | **Course Title** | **Research Methodology** |
| **No of Credits** | **2 Credits** (TH) | **Internal** | **20%** |
| **Total Contact Hrs./Week** | **2 HRS** (TH/Week) | **External** (Semester end Exam.) | **80%** |

### Course Outcomes:

- Critically analyze research methodologies identified in existing literature.
- Choose appropriate quantitative or qualitative method to collect data.
- Propose and distinguish appropriate research designs and methodologies to apply to a specific research project.
- Develop a comprehensive research methodology for a research question.
- Apply the understanding of feasibility and practicality of research methodology for a proposed project.

**Course Outline:**

**Unit 1: Definition of research:** Research – Definition; Concept of Construct, Postulate, Proposition, Thesis, Hypothesis, Law, Principle. Definition and Dimension of a Theory, Functions and Characteristics; Types of Theory: General Theory and Particular/ Empirical Theory. Cases and their Limitations; Causal Relations. Philosophy and validity of research. Objective of research.

**Unit 2: Characteristics of research:** Various functions that describe characteristics of research such as systematic, valid, verifiable, empirical and critical approach.

**Unit 3: Types of research:** Pure and applied research. Descriptive and explanatory research, Qualitative and quantitative approaches.

**Unit 4: Research procedure:** Formulating the Research Problem, Literature Review, Developing the objectives, Preparing the research design including sample Design, Sample size.

**Unit 5: Considerations in selecting research problem:** Relevance, interest, available data, choice of data, Analysis of data, Generalization and interpretation of analysis.

**Unit 6: Outcome of research:** Preparation of the Report on conclusions reached. Testing validity of research outcomes. Suggestions and recommendations, identifying future scope.

**Reference Books:**


* Dawson, Catherine, 2002, *Practical Research Methods,* New Delhi, UBS Publishers' Distributors.
* Kothari, C.R.,1985, *Research Methodology-Methods and Techniques,* New Delhi, Wiley Eastern Limited.

* Kumar, Ranjit, 2005, *Research Methodology-A Step-by-Step Guide for Beginners*, (2nd.ed), Singapore, Pearson Education.

## 2. Advance Operating System

| | | | |
|---|---|---|---|
| **Course Code** | **CSC403** | **Course Title** | **Advance Operating System** |
| **Number of Credits** | **3 Credits** (TH)<br>**2 Credits** (PR) | **Internal** | **20** |
| **Total Contact Hours** | **3 HRS** (TH/Week)<br>**4 HRS** (PR/Week) | **External** (Semester/Term Exam) | **80** |

## Course Outcomes:

- Gain extensive knowledge on principles and modules of operating systems
- Understand key mechanisms in design of operating systems modules
- Understand process management, concurrent processes and threads,memory management, virtual memory concepts, deadlocks.
- Compare performance of processor scheduling algorithms.
- Produce algorithmic solutions to process synchronization problems.
- Use modern operating system calls such as Linux process and synchronization libraries.
- Learn thread and multicore programming
- This course introduces basic issues in operating systems. Topics include Threads, processes, concurrency, memory management, I/O Control and case studies.

### Course Outline

**Unit - 1: Introduction**

Introduction to hardware support for operating systems: privileged mode execution, saving and restoring CPU state, traps and interrupts, timers, memory protection. Operating system techniques for protecting user and hardware resources. Overview of the key operating system abstractions and the use of system calls to manipulate them. Definition of OS, Types of OS: main Frame , Desktop, Multiprocessor, Distributed, Clustered, Real time, Multi programming, Time Sharing, Embedded OS . Overview of operating systems responsibilities. Operating system components and structures. Desirable Qualities of OS. **Process :** Definition Processes and programs. Implicit and Explicit tasking, Program execution, Independent and Co-operation process, Process relationship, Process - related states, Process State transitions, Process Control Block, Context switches, Threads: Necessity and Advantage of Threads, Types of Threads. System Calls and System call Execution. **Interprocess Communication:** Basic concepts ,Shared Memory System, Message Passing: Direct versus Indirect Communication, Critical Sections, Race conditions, Mutual Exclusion. **Process Scheduling:** Objectives of scheduling, Types of Schedulers: Short, Long, Medium, Comparision between schedulers, Scheduling Criteria, Scheduling Algorithms: Types Preemptive and NonPreemptive scheduling, FCFS, RR, SJF and Priority based Scheduling; Evaluation of Scheduling algorithms.

**Unit - 2:**

**Threads and Concurrency:** Threads, process context switch vs thread switch, true concurrency vs pseudo concurrency, operating systems as concurrent programs, concurrency

through multi-threading, concurrency through interrupt handling, concurrent access to shared memory, race conditions, mutual exclusion, synchronization primitives based on atomic instructions. Thread programming using OpenMP: OpenMP programming model,Specifying current task in OpenMP, Synchronization Conctructs in OpenMP, Data Handling, Library function, Environment variables. **Synchronization Primitives:** Atomic instructions, locks, spinlocks, mutex semaphores, counting semaphores, and their use in solutions to Producer Consumer synchronization. **Classic Synchronization Problems:** Classic synchronization problems: Producer Consumer, Dining Philosophers, Readers and Writers, Sleeping Barber. **Monitors and Message Passing:** Monitors, condition variables, message passing, and their use in solutions to classic synchronization problems: Producer Consumer, Dining Philosophers, Readers and Writers, Sleeping Barber. ' **Deadlock** Deadlock: Definition, Characteristics A resource Allocation graph, livelock, Deadlock prevention, Deadlock avoidance: Banker s Algorithm, Deadlock Detection and Recovery.

**Unit - 3:**

**Introduction:** Overlays and Swapping, Logical and Physical address space ,Contiguous allocation methods, Single partition and multiple partition Systems, Relocation Memory Mangement. **Paging:** Principle of operation, Page allocation, Hardware support for paging, Paging address translation by direct mapping and associate mapping , Protection and sharing, Advantages and disadvantages of paging **Segmentation:** Principle of operation, Address translation, Advantages and disadvantages of segmentation , Comparison between segmentation and Paging. **Virtual Memory:** Necessity, Hardware and control structures, Locality of reference, Page fault, Working set, Dirty page/Dirty bit, Demand paging, Thrashing, Page replacement Policies: FIFO, LRU, NRU.

**Unit - 4:**

**I/O Management:** I/O buffering ,Single and Double Buffer schemes, Disk Organization. **File Management:** File Concepts, File descriptor, Access methods: Sequential, indexed and direct, File sharing, Protection, Access rights, File System structure, Byte Sequence, Record sequence and Tree-based, Recovery and Disk formatting. **Secondary Storage Management:** File allocation methods: Contiguous Allocation and Non Contiguous allocation , Chained, Indexed allocation , free space management , Disk Scheduling: FCFS, SSTF, SCAN and C-SCAN , Disk Cache. **Protection and Security:** System performance, protection and security, policies and methods, Access Matrix.

**Unit - 5:**

**Introduction :** History of Linux , Features of Linux, Drawbacks of Linux , Components of Linux, Memory Management Subsystems , Linux Process and Thread Management, File Management System, Device Drivers **Linux Commands and Utilities:** cat, tail, cmp, diff, wc, sort , mkdir, cd, rmdir, pwd, cp, more, passwd, who, whoami, mv, chmod, kill, write, wall, merge , mail, pipes, filters and redirection utilities. **Shell Scripts:** Creating and executing simple shell programs, variables, special characters, comparison of expressions, iteration statements, conditional statements functions. **System Administration:** Installing Linux, Booting the system, Maintaining user accounts, File systems and Special Files, Backups and Restoration.

**Books:**

i) Operating Systems Concepts, 8th edition, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne; Wiley, ISBN 0-470-12872-0,2010.

**Reference Books:**

ii) Operating Systems: Internals and Design Principles, 6th edition, William Stallings; Prentice Hall, ISBN-10: 0136006329, Operating Systems, 3rd edition

iii) Modern Operating Systems, Andrew S. Tanenbaum; Prentice Hall, ISBN-10: 0-13-600663-9, 2008, 3rd edition.

iv) Using Open MP, Portable Shared Memory Parallel Programming ,Barbara Chapman, Gabriele Jost and Ruudvander Pas, MIT Press, ISBN: 9780262533027 ,2007

**Website:**

(i) http://codex.cs.yale.edu/avi/os-book/OS8/os8c/slide-dir

(ii) http://openmp.org/wp/resources/

(iii)http://www.compunity.org/training/tutorials/3%20Overview_OpenMP.pdf

**Lab Exercise: CSC451 Practical based on CSC403** At least five experiments should be carried out on each unit.

By considering the sample experiment list as guidelines :

- Implementing a CPU scheduling policy in a Linux OS.
- Implementing a memory management policy in a Linux OS.
- Implementing a file system in a Linux OS.
- Process control system calls: The demonstration of fork, execve and wait system calls along with zombie and orphan states.
- Thread management using OpenMP API.: Thread execution ,Static scheduling, Dynamic scheduling , Synchronization Conctructs in OpenMP ,Data Handling,Library function,Environment variables.
- Thread synchronization using counting semaphores and mutual exclusion using mutex. Application to demonstrate: producer-consumer problem with counting semaphores and mutex.
- Deadlock Avoidance Using Semaphores
- Implement the deadlock-free solution to Dining Philosophers problem to illustrate the problem of deadlock and/or starvation that can occur when many synchronized threads are competing for limited resources.
- Demonstrate the following CPU Scheduling Algorithms

  a. FCFS b. SJF c. Priority d. Round Robin

T Demonstrate all Page Replacement Algorithms

  a. FIFO b. LRU c. MRU

- Simulate Bankers algorithm for Deadlock Avoidance
- Simulate Bankers Algorithm for deadlock Prevention

## 3. Data Structure and Algorithm

| | | | |
|---|---|---|---|
| **Course Code** | | | **Algorithm** |
| **No of Credits** | **3 Credits** (TH)<br>**4 Credits** (PR) | **Internal** | **20%** |
| **Total Contact Hrs./Week** | **3 HRS** (TH/Week)<br>**4 HRS** (PR/Week) | **External** (Semester end Exam.) | **80%** |

**Course Outcomes:**

- Understand structure and behavior of Algorithms.

- Better scope to write effective programs.
- Helpful in the preparation of UGC SET/NET, various entry level Examinations.

**Course Outline:**

**Unit-1: Introduction to Data Structure & Algorithm Notations**

Introduction to Data Structure, Types of data structure 1. Primitive 2. Non Primitive 3. Linear 4. Nonlinear, Need of data structure, Algorithm Notations, Format Convention, Name of Algorithm, Introductory Comment, Steps, Data Structure, Arrays, Dynamic Storage allocation, Functions Procedures

**Unit-2: Introduction to Algorithm analysis for Time and Space Requirement** Rate and Growth, Basic time analysis of an algorithm, Order Notation, More timing Analysis, Space analysis of an algorithm. Arrays□ Ordered List, Sparse Metrics, Array Representations

**Unit-3:Elementary Stacks Data and Structures: Queues:-Organization, Operations, multiple stacks, types of** queues, **Linked Lists:- Singly Linked List, Doubly Linked List, Doubly Linked List and** Dynamic Storage management, Garbage collection and Compaction

**Unit-4: Trees:** Terminology, Binary Trees, Tree representation, Tree traversal, Threaded Binary Trees, Binary Search trees, B Tree, **Graph Algorithms:** Terminology and Representation,' **Traversal:** BFS, DFS, Connected Components and Spanning trees, Dijkstra s algorithm.

**Unit-5: Searching and Sorting: Linear Vs Binary search, Sorting: - Insertion Sort, Merge** Sort, Quick sort, Radix Sort.

**Text Books:**

1. Fundamentals of Data Structures by Ellis Horowitz and Sartaj Sahani.
2. Introduction to Algorithms by Thomas H Core man et.al, PHI Publication.

**References :**

- An introduction to data structures with applications, Jean-Paul Trembley, Paul. G. Soresan, Mc-Graw Hill International Editions
- Data Structures, Howorithiz, Sahani, Galgotia publication
- Data Structures and Algorithms, Aho, Hopcroft, Ulman,
- Data Structures using C and C++, Tannenbaum, PHI.

**E-Books:**

**http:**//www.tutorialspoint.com//java/java_data_structures.htm

**Lab Exercise: CSC452 Practical based on CSC404:**

Practical (Lab Work) :

**(a)** Students should develop various algorithms in Java .
**(b)** Analyse the programs in terms of space and time complexity.
**(c)** Use these programs for further development in their projects.

## 4. Discrete Mathematical Structure

| | | | |
|---|---|---|---|
| **Course Code** | **CSC405** | **Course Title** | **Discrete Mathematical Structure** |
| **Number of Credits** | **3 Credits** (TH)<br>**2 Credits** (PR) | **Internal** | **20%** |
| **Total Contact Hours** | **3 HRS** (TH/Week)<br>**4 HRS** (PR/Week) | **External** (Semester/Term Exam) | **80%** |

### Course Outcomes:

- An in depth knowledge on various discrete structures available.
- Understand the theory and techniques of logic, graphs and trees, and algebraic systems
- Apply the knowledge and skills obtained to investigate and solve a variety of discrete mathematical problems
- Communicate mathematical ideas
- Gaining of some confidence on how to deal with problems which may arrive in computer science in near future.
- Solve problems using counting techniques and combinatorics.
- Perform operations on discrete structures such as sets, functions, relations or sequences.
- Solve problems involving recurrence relations and generating functions.
- Construct functions and apply counting techniques on sets in the context of discrete probability.
- Apply algorithms and use definitions to solve problems to proof statements in elementary number theory

### Course Outline

**Unit - 1:**

**Introduction to Logic:** Basic definitions and notation, Appropriate use of quantifiers, Tautologies and contradictions. **Sets:** Notation, operations and relations, Finite and infinite sets, Principle of Inclusion & Exclusion. **Relations and Functions:** Basic definitions and properties, Binary relations, Equivalence relations and partitions.

**Unit - 2:**

**Mathematical Induction:** The Well Ordering Principle, Proof by mathematical induction **Recursion:** Recursively defined sequences, Linear recurrence relations with constant coefficient, **Algorithms:** Basic concept of algorithms, Analysis of algorithms, Euclidean algorithm, Searching and sorting algorithms

**Unit - 3:**

**Counting Techniques:** Fundamental counting techniques,' Permutations and combinations, The Pigeonhole Principle, Binomial coefficients and Pascal s Triangle, Introduction to generating functions.

**Unit - 4:**

**Graph Theory:** Fundamental concepts of graphs and subgraphs, Weighted graphs, Paths and circuits, Euler and Hamiltonian paths and circuits, Planar graphs, Graph coloring. **Trees:** Basic definitions and properties of trees, Spanning trees, Weighted trees

**Unit - 5:**

**Boolean Algebra:** Boolean Functions, Representing Boolean Functions, Logic Gates, Minimization of Circuits, **Modeling Computation:** Languages and Grammars, Finite-State Machines with Output, Finite-State Machines with No Output, Language Recognition, Turing Machines.

**Books:**

- Discrete Mathematics and Its Applications, Kenneth H. Rosen, Mc Graw Hill Education,
  ISBN-13: 978-0072899054, 4th Revised edition edition,1999

**Website:**

- **https://people.cs.pitt.edu/~milos/courses/cs441/**

**Lab Exercise: CSC453 Practical based on CSC405:** At least two experiments should be carried out on each unit.

**Practical Assignments for DMS:**

Practical can be performed in Python or any other programming language, following is the list of some assignments:

Q1. Write a program to find BCD using Euclidian Algorithm of two numbers.
Q2. Write a program for linear search using array.
Q3. Write a program for binary search using array.
Q4. Write a program to find factors of two number.
Q5. Write a program for union of two sets using array.
Q6. Write a program for intersection using array.
Q7. Write a program forset difference using array.
Q8. Write a program for recursion using array.
Q9.How probability density function works. Elaborate.
Q10. WAP to define sets in along with its operations.

Along with these assignments, more programs can be developed that will use Problem Solving techniques in Discrete Mathematical Structures.

## 5. Programming in Core Java

| | | | |
|---|---|---|---|
| **Course Code** | **CSC406** | **Course Title** | **Programming in Core Java** |
| **Number of Credits** | **3 Credits** (TH)<br>**2 Credits** (PR) | **Internal** | **20%** |
| **Total Contact Hours** | **3 HRS** (TH/Week)<br>**4 HRS** (PR/Week) | **External** (Semester/Term Exam) | **80%** |

## Course Outcomes:

- Good JAVA Programmer.
- Useful for NET/SET Examination.
- Useful for JAVA Certification.
- Useful for Applets and Apps Development.

**Course Outline**

**Unit-1:**Program, The main() Method, Useful Stuff Necessary to go Further, System.out.println(), Using the Java Documentation. **B. Java Basics:** Basic Java Syntax: General Syntax Rules, Java Statements, Blocks of Code, Comments, Variables: Data types, Primitive Data Types, Object Data Types, Literal Values, Constants and the final keyword, Mathematics in Java: Expressions, Operator Precedence, Multiple Assignments, Order of Evaluation, Bitwise operators, Compound Operators, Expressions that Mix Data Types: Typecasting Creating and Using Methods, Creating Methods, Variable Scope.

**Unit-2:Java Objects**: **Objects: Object-Oriented Languages, Object-Oriented Programs, Encapsulation,** Creating and Using an Instance of an Object, References Defining a Class, Constructors, Method Overloading, The this Keyword, static Elements, Garbage Collection, Java Packages, Dealing with Controlling Keyboard Input, Program String, Flow: String Buffer, Boolean Valued and String Builder, Expressions, Complex Creating Boolean Documentation, Expressions, Comments Simple Branching, and Using Two java doc, Mutually Java doc Exclusive Comments Branches,.. Nested Comparisons if...else Statements And Flow Comparing ntol Structures a Number: of Mutually Exclusive Options, Comparing a Number of Mutually Exclusive Option, The switch Statement, Comparing Objects, Conditional Expression, while and do. . .while Loops, for Loops, Additional Loop Control: break and continue, Breaking Out of a Loop, Continuing a Loop, Classpath, Code Libraries, and JAR files, Using CLASSPATH Creating a jar File (a Library) C. Loops Arrays and the Vectors: For, Each Arrays Loop, : Multi Defining Dimension ald Declaring Arrays, Arrays, Multidimensional Instaating Arrays in Initializing Memory, Example Arrays, Work Printing With a Picture, Arrays Array Type casting Variables, with Copying Arrays Arrays,of Primitives,Arrays Using Objects, Vectors: Enhanced Defining for Vectors and using Vectors.

**Unit-3: Constructors** Inheritance:the Inheritance:super Keyword, Derived Derived Class Objects,Class Methods Polymorphism, Override Inheritance Base and Class References Methods Dynamic Method Invocation, Creating a Derived Class, Inheritance and Access Inheritance and Inheritance and Default Base Class Constructors, The Instantiation Process at Runtime, Typecasting with Object References: Typecasting, Polymorphism, and Dynamic Method Invocation, More on Overriding, Object Typecasting Example, Checking an Object's Type: Using instanceof, Typecasting with Arrays of Objects, Other Inheritance□Related Keywords: abstract, final, Methods Inherited from Object. B. Packages and Interfaces: Interfaces: Creating an Definition, Implementing Interfaces: Implementing Interfaces Example, Reference Variables and Calling an I terface Method, Interfaces and Inheritanc : Some Use for Interfaces, Interfaces ,and Event Handling Interfaces and "Pluggable Components", Packages: Creating and using packages, Access. **C. Inner Classes:** Inner Classes, Nested Classes, Inner Class Syntax, Instantiating an Inner Class Instance from Within theng Enclosing Risky Code Class, try Inner and Classes catch, Guaranteeing Referencd from Execution Outside of the Codc Enclosing the finally Class Block,Working Letting with an Inner Exception Classes be. Thrown D.Exceptions:to the Exceptions: Handling Excepti ns, Exception Objects: At empty Method Caller, Throwing an Exception, Exceptions and Inheritance, Exception Class, Constructors and Methods, Creating and Using Your Own Exception Classes, Rethrowing Exceptions, Initializer Blocks, Static Initializer Blocks, Assertions.

**Unit-4:**

**A.Utility Classes**: Collection Interfaces, Concrete collections, Collections framework, Algorithms, Legacy Collections Streams: Output Streams, Input Streams, Filter Streams,

Readers and Writers. **B. Threads**: Thread Class and Runnable Interface, Thread Synchronization **C. I/O Package**: InputStream and OutputStream classes, Reader and Writer classes **D. Java Networking:** InetAddress, URL, URLConnection, TCP/IP Server Socket, Client Socket, User Datagram Sockets, **D. Applet and Swings**: Applet: Applet Life Cycle, Passing Parameters to Applet, Delegation Event Model, AWT Components, AWT Events, using listeners, Working with Graphics, Loading Image and Multimedia objects in applet

**Unit-5: Java Database Connectivity**: Java Database Connectivity Architecture, JDBC-ODBC Bridge, JDBC Drivers, JDBC API, JDBC classes, Driver Interface, DriverManager Class, Connection, Statement, ResultSet, Implementing Stored Procedures.

**Reference Books**:

1) Java 2 Complete Reference by Herbert Schieldt (Sixth Edition)
2) Core Java Vol 1: Sun Press, Eighth Edition
3) Core Java Vol 2: Sun Press

**E-books**:

- Java 2 Complete Reference by Herbert Schieldt (Fourth Edition)

**Lab Exercise for CSC454, Practical based on CSC406:**There should be minimum 20 lab assignment on the topics discussed in the course using Open Source Platform (Eclipse, Net Bean etc.).

## *Semester – II*

## 1. Research Project Review Writing

| | | | |
|---|---|---|---|
| **Course Code** | **CSC407** | **Course Title** | **Research Project Review Writing** |
| **No of Credits** | **1 Credits** (TH) | **Internal** | **100%** |
| **Total Contact Hrs./Week** | **1 HRS** (TH/Week) | **External** (Semester end Exam.) | - |

## Course Outcomes:

- With this course, students will become familiar with and learn to identify the most relevant textbooks, reviews, papers and journals for their research topics. During the course the students will also learn how to critically read and assess research papers and reviews. The review should point to research gaps that can be operationalized into feasible research questions.

**Course Guidelines:**

**1. What is a literature review?**

The ability to review, and to report on relevant literature is a key academic skill. A literature review: situates your research focus within the context of the wider academic community in your field; reports your critical review of the relevant literature; and identifies a gap within that literature that your research will attempt to address.

**1. Why do I need a literature review?**

When readers come to your assignment, dissertation, or thesis, they will not just assume that your research or analysis is a good idea; they will want to be persuaded that it is relevant and that it was worth doing. They will ask questions such as: What research question(s) are you asking? Why are you asking it/them? Has anyone else done anything similar? Is your research relevant to research/practice/theory in your field? What is already known or understood about this topic? How might your research add to this understanding, or challenge existing theories and beliefs? These are questions that you will already probably be asking yourself.

**3) A critical review**

It is important that the literature review is more than just a list of references with a short description of each one. Merriam (1988:6) describes the literature review as: 'an interpretation and synthesis of published work'.

This very short statement contains some key concepts, which are examined in the table below.

A] Published work 's Concepts Explanation and its Associatcd critique B]Interpretation.

C]Synthesis

**1. Getting started**

Reading anything on your research area is a good start.

According to Taylor and Procter (2008) of The University of Toronto have some useful suggested questions to ask yourself at the beginning of your reading:

1.What is the specific thesis, problem, or research question that my literature review helps to define?

2.What type of literature review am I conducting?

3.Am I looking at issues of theory? methodology? policy? quantitative research? qualitative research?

4.What is the scope of my literature review?

5.What types of publications am I using (e.g., journals, books, government documents, popular media)?

6.What discipline(s) am I working in (e.g., GIS,Biometrics ,Imageprocessing ,NLP)?

- **Ways of finding relevant material**

  Electronic sources.

  References of references

  Hand searching of journals
- **Collecting material**
- **Keeping a record**
- **Plagiarism Detection**
- **When to stop**

It is important to keep control of the reading process, and to keep your research focus in mind. Rudestam and Newton (1992:49) remind us that the aim is to 'Build an argument, not a library'. It is also important to see the writing stage as part of the research process, not something that happens after you have finished reading the literature.

**1) Writing it up**

The task of shaping a logical and effective report of a literature review is undeniably challenging. Some useful guidance on how to approach the writing up is given by Wellington et al (2005:87): "It should be framed by your research questions. It must relate to your study. It must be clear to the reader where it is going: keep signposting along the way. Wherever possible, use original source material rather than summaries or reviews by others.

**1. Using tables**

As well as using tables to display numerical data, tables can be useful within a literature review when you are comparing other kinds of material.

**1. Reference list**

Almost all academic writing will need a reference list. This is a comprehensive list of the full references of sources that you have referred to in your writing.

**References**

1. Ask Oxford (2006). Found at: http://www.askoxford.com/

**Useful reading**

1) Murray, R. 2003: How to survive your viva. Maidenhead: Open University Press.

2) Rugg, G. & Petre, M. 2004: The unwritten rules of PhD research. Maidenhead: Open University Press.

1. Tinkler, P. & Jackson, C. 2004: The doctoral examination process: a handbook for students, examiners and supervisors. The Society for Research into Higher Education. Maidenhead: Open University Press.

2. Wellington, J., Bathmaker, A., Hunt, C., McCulloch, G. & Sikes, P. 2005: Succeeding with your doctorate. London: Sage.

## 2 Relational Database Management System and SQL

| | | | |
|---|---|---|---|
| **Course Code** | **CSC408** | **Course Title** | **Relational Database Management System and SQL** |
| **Number of Credits** | **3 Credits** (TH)<br>**2 Credits** (PR) | **Internal** | **20%** |
| **Total Contact Hours** | **3 HRS** (TH/Week)<br>**4 IIRS** (PR/Week) | **External** (Semester/ Term Exam) | **80%** |

### Course Outcomes:

- Understand, appreciate and effectively explain the underlying concepts of database technologies
- Design and implement a database schema for a given problem-domain
- Normalize a database
- Populate and query a database using SQL DML/DDL commands.
- Declare and enforce integrity constraints on a database using a state-of-the-art RDBMS
- Programming PL/SQL including stored procedures, stored functions, cursors, packages.
- Design and build a GUI application using a 4GL

### Course Outline

**Unit-1:**

**Introduction:** An overview to DBMS, Data models, levels of abstraction, data independence, instances and schema, structure of DBMS, database users, functions of database administrators. **Entity relationship model:** Entities, Attributes and Entity sets, Relation and relationship sets, features of E-R model.

**Unit–2:**

**Relational model:** Introduction, Integrity constraints over relations, Enforcing data integrity, relational data, logical database design, introduction to views,

**Unit-3:**

Relational algebra and relational calculus: operations on rclational algebra, operations on relational calculus, tuple relational calculus, domain relational calculus

**Unit-4:**

Relational Database Design: Functional dependencies, schema refinement, Normal forms- first, second, third, BCNF, fourth and fifth normal forms, multi-valued dependencies. Structured Query language(SQL): Basic SQL queries, nested queries, aggregate operators, null values, Embedded SQL. Query Processing: Concept, Need of query processing, procedure and its importance , query optimization, estimation of query processing cost, structure of query optimizer and Join strategies. Indexing and Hashing:-Basic concepts, B+ Tree Index files, B-tree index files, Static Hash functions, Dynamic Hash functions, Comparison of Indexing and Hashing, Index definition in SQL.

**Unit-5:**

Internals of RDBMS: Transaction management, ACID property, Schedules and its various types, serializability, deadlock handling, high performance transaction systems. Concurrency Control: Lock based protocols, Timestamp based protocols, Validation techniques, Multiple Granularity. Crash Recovery:-Failure classification, Log based recovery, Checkpoints, Shadow paging. Distributed Databases:-Structure and design of distributed databases, Design of distributed databases, Transparency and autonomy, distributed query processing, recovery in distributed databases, Commit protocols. Security and Integrity: Authorization and views, security specification in SQL, Encryption.

**Books:-**
Database System Concepts by Henry Korth and Abraham Silberschatz

**Reference Books:-**
Relational database systems by Desai, Relational database concepts by Aho Ullman.

**Lab Exercise:-**
**CSI455 Practical based on CSC408:**
At least two practical's should be carried on each unit.

## 3 Software Engineering and CASE Tools

| | | | |
|---|---|---|---|
| **Course Code** | **CSC409** | **Course Title** | **Software Engineering and CASE Tools** |
| **Number of Credits** | **3 Credits** (TH)<br>**2 Credits** (PR) | **Internal** | **20%** |
| **Total Contact Hours** | **3 HRS** (TH/Week)<br>**4 HRS** (PR/Week) | **External** (Semester/Term Exam) | **80%** |

### Course Outcomes

- Plan and deliver an effective software engineering process, based on knowledge of widely used development lifecycle models.
- Employ group working skills including general organization, planning and time management and inter-group negotiation.
- Translate a requirements specification into an implementable design, following a structured and organised process.
- Formulate a testing strategy for a software system, employing techniques such as unit testing, test driven development and functional testing.
- Evaluate the quality of the requirements, analysis and design work done during the module.
- Apply new software models, techniques and technologies to bring out innovative and novelistic solutions for the growth of the society in all aspects and evolving into their continuous professional development.

### Course Outline

The Nature of Software, Defining Software, Legacy Software, Software Engineering, Software characteristics, Application software. So waterfall model, incremental and evolutionary models. Requirement engineering: Requirement engineering task, initiating the requirement engineering process, eliciting requirements, developing use cases, building analysis model, Negotiating requirements, validating requirements, data modeling, functional

modeling and behavioral modeling. Requirements Specification: Characteristics of an SRS, Components of an SRS, Specification Languages, Structure of a requirements document.

**Unit-2:**

Design within the Context of Software Engineering, The Design Process, Design Concepts, The Design Model. **Architectural Design**: Software Architecture, Architectural Genres, Architectural Styles Architectural Design, Assessing Alternative Architectural Designs. **Component Level Design:** Designing Class-Based Components, Conducting Component-Level Design, Cohesion and Coupling. **User Interface Design**: The Golden Rules, Interface Analysis and Design, Interface Analysis Interface Design Steps.

**Unit-3:**

Verification and Validation, Testing Overview: Verification vs Validation, Design of test cases - Box testing: Equivalence Class Partitioning, Graph based testing Boundary Value analysis White- Box Testing: Statement Coverage, Branch Coverage, Condition Coverage, Path Coverage, Cycloramic Complexity Metric Data Plow- Based Testing. Integration Testing: Top down Testing, Bottom Up testing, Regression Testing, fazed vs Incremental Integration testing Systems Testing: Stress Testing Recovery Testing Security Testing. Debugging Techniques, Approaches, Tools

**Unit-4:**

Project Management Concepts, Process and project– metrics, Process Metrics and Software Process Improvement, Project Metrics, Software Measurement, Size-Oriented Metrics, Function-Oriented Metrics, Object-Oriented Metrics, Use-Case Oriented Metrics. Project Planning Process: Software Scope and Feasibility, Resources, Software Project Estimation, Decomposition Techniques, Empirical Estimation Models, Scheduling, Risk analysis, monitoring and management. Software Configuration Management, Quality Management.

**Unit - 5:**

Clean room Software Engineering, Web Engineering, Software Reengineering, Reverse Engineering, Forward Engineering, Computer Aided Software Engineering.

**Books:**

1 Software Engineering, A Practitioners Approach Roger S. Pressman, 4th /7th Edition,
2 An Integrated Approach– To S/w Engineering, Pankaj Jolote, 1st / 2nd Edition, Narosa.
3 Software Engineering A Programming Approach, D. Belie I. Moray, J. Rough, PHI. TataMc GrawHill,International Education.

**Reference Books:**

2) James Peter, W Pedrycz, Software Engineering , John Wiley & Sons
3) K K Aggarwal & Yogesh Singh, Software Engineering , New Age International, 2001

**E-books:**

Software Engineering – A Practitioners Approach Roger S. Pressman, 5th Edition, Tata McGraw Hill, International Education.

**Lab Exercise:**
**CSI456 Practical based on CSC409:**

The practical part, students will allotted a case study /Mini Project development of software products from an industry perspective, including generation of appropriate documents, under tight schedules and limited resources.

## 4 Compiler Design

| | | | |
|---|---|---|---|
| **Course Code** | **CSC410** | **Course Title** | **Compiler Design** |
| **Number of Credits** | **3 Credits** (TH)<br>**2 Credits** (PR) | **Internal** | **20%** |
| **Total Contact Hours** | **3 HRS** (TH/Week)<br>**4 HRS** (PR/Week) | **External** (Semester/Term Exam) | **80%** |

### Course Outcomes:

- Fluency in describing the theory and practice of compilation, in particular, the lexical analysis, syntax, and semantic analysis, code generation and optimization phases of compilation.
- Ability to create lexical rules and grammars for a programming language.

**Course Outline**

**Unit-1: Introduction**

Translator Issues, why to write a Compiler, what is a Compiler, what is the Challenge, Compiler Architecture, Front end and Back end model of compiler, Cross compiler, Incremental compiler, Boot strapping, Lexical Analysis: Concept of Lexical Analysis, Regular Expressions, Deterministic finite automata (DFA), Non-Deterministic finite automata (NFA), Converting regular expressions to DFA, Converting NFA to DFA, Hand coding of Lexical analyzer, Introduction to LEX Tool and LEX file specification, Error detection and recovery in LEX.

**Unit-2: Syntax Analysis**

Context Free Grammars(CFG), Concept of parsing, Parsing Techniques, Top-Down Parsers: Introduction, Predictive Parsing - Removal of left recursion, Removal of left factoring, Recursive Descent Parsing, Predictive LL(k) Parsing Using Tables, Bottom Up parsing: Introduction, Shift-Reduce Parsing Using the ACTION/GOTO Tables, Table Construction, SLR(1), LR(1) and LALR(1) Grammars, Practical Considerations for LALR(1) Grammars, Introduction to YACC Tool & YACC file specification, Error detection and recovery in YACC.

**Unit-3:**

**Semantic Analysis & Intermediate Representation:** Need of semantic analysis, Abstract Parse trees for Expressions, variables, statements, functions and class declarations, Syntax directed definitions, Syntax directed translation schemes for declaration processing, type analysis, scope analysis, Symbol Tables (ST),Organization of ST for block structure and non-block structured languages, Symbol Table management, Type Checkers: type checking for expressions, declarations (variable, type, function, recursive), statements, Intermediate code generation: Intermediate languages, Design issues, Intermediate representations: three address, postfix & abstract syntax trees, Intermediate code generation for declaration, assignment, iterative statements, case statements, arrays, structures, conditional statements, Boolean expressions, procedure/function definition and call.

**Unit-4:**

**Run-Time Memory Management & Code generation:**

Model of a program in execution, Stack and static allocation, Activation records, Issues in the design of code generation, Target machine description, Basic blocks & flow graphs, Expression Trees, Unified algorithms for instruction selection and code generation, Sethi Ullman algorithm for expression trees, Aho Johnson algorithm, Different models of machines , order of evaluation, register allocation, Code generator-generator concept .

**Text Books:**

3 Alfred V. Aho, A. V. R. Sethi and J.D. Ullman Compiler Principle, Techniques and Tools Addison Wesley.

**Reference Books:**

- Barrent W. A., J. D. Couch, Compiler Construction Theory and Practice , Computer Science series, Asian student edition.
- Dhamdhere D.M., Compiler Construction Principle_ and Practice , Mac. Millan India, New Delhi.
- Manish Kumar Jhas, Compiler Construction An advance course .
- John Levine, Tony Mason & Doug Brown, Lex and Yacc
- Ravendra Singh, Vivek Sharma, Manish Varshney, Design and Implementation of Compiler , New Ag Publications. Prentice Hall.**O**, Reilly

**Lab Exercise: CSC457 Practical based on CSC410:**
**List of Experiments:**

1.Design a lexical analyzer for given language and the lexical analyzer should ignore redundant spaces, tabs and new lines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value. Simulate the same in C language.
2.Write a C program to identify whether a given line is a comment or not .
3.Write a C program to test whether a given identifier is valid or not.
4.Write a C program to simulate lexical analyzer for validating operators .
5.To Study about Lexical Analyzer Generator(LEX)
6.Create a Lexer to take input from text file and count no of characters, no. of lines & no. of words.
7.Write a Lex program to count number of vowels and consonants in a given input string.
8.Write a program which adds line numbers to the given file and display the same onto the standard output.
9.Write a Lex program to count the number of comment lines in a given C program. Also eliminate them and copy that program into separate file.
10.Write a C program for implementing the functionalities of predictive parser for the mini language.
11.Write a C program for constructing of LL (1) parsing.
12.Write a C program for constructing recursive descent parsing
13.Write a C program to implement LALR parsing.
14.Write a C program to implement operator precedence parsing.

## 5 Advanced Java

| | | | |
|---|---|---|---|
| **Course Code** | **CSC411** | **Course Title** | **Advance Java** |
| **Number of Credits** | **3 Credits** (TH)<br>**2 Credits** (PR) | **Internal** | **20%** |
| **Total Contact Hours** | **3 HRS** (TH/Week) | **External** | **80%** |

4 HRS (PR/Week) (Semester/ Term Exam)

## Course Outcomes:

- Students can appear for java certification examinations. Student can also work on networking and web Projects.

**Course Outline**

**Unit-1:**

**Java Server Pages**: Basics and Overview, JSP architecture, JSP tags and JSP expressions, Fixed Template Data, Lifecycle of a JSP, Model View Controller (MVC), Model 1/Model 2 Architecture, Data Sharing among servlets & JSP: Object scopes or "buckets", Request, application, session and page scope, Predefined JSP implicit objects (request, session, application, page), <jsp:useBean>, <jsp:getProperty>, <jsp:setProperty>, <jsp:include>, <jsp:forward>, More JSP Capabilities and Session Management, HTTP as a stateless protocol, Hidden form fields, Cookies: Overview, API, Using cookies, Session overview: Cookies and session tracking, HttpSession, Putting data into a session object, Retrieving data from a session object, Using session data in servlets and JSPs Additional JSP Capabilities, Exception handling and error pages, Directives (page, include, others), Import declarations, Multithreading considerations and data safety, SingleThreadModel interface, Additional JSP Capabilities, JSP Directives, JSP Error Pages, JSP and Java Declarations, Scriptlet overview, Scriptlet syntax

**Unit-2:**

**JSTL:** Using Custom Tags, Custom tags overview, Reducing JSP complexity, Tag Libraries, Tag Library Descriptor (TLD), Loading a tag library in a web app, The JSTL, JSP Expression Language (EL), Using custom tags, The c:url, c:param, c:forEach, c:out tags, Overview of JSTL libraries, The JSTL Expression Language, Expressions, Type Coercion, Operators, String concatenation, Implicit Objects, The Core JSTL Library, General Purpose: c:out, c:set, c:catch, Conditional: c:if, c:choose,, Overview of other– capabilities, Additional Topics : Servlet Filter overview, Filtering examples, lifecycle, & filter chains, Filter API, Modifying a request, Modifying a response, Struts Overview Advanced MVC Struts overview, Command and State patterns, Struts View and Controller elements

**Unit-3:**

**API:Servlets:**-HTML Web **Forms,** Application **HTTP:Request** Basics:- **response,** How the Web **headers,** works, **GET,**Thin **POST,** Clients, **Overview:** TCP/IP, **How** HTTP **Servlets** overview, **Work,** Brief review, Overvi w of Java EE, servlets & Web applications., Servlet Basics, Servlet Servlet Lifecycle: init(), service(), destroy(), Requests and responses, Core Servlet API: Generic Servlet, Servlet Request, and Servlet Response, HTTP Servlets: Http Servlet Request, Http Servlet Response and Http Servlet, Accessing Parameters, Additional Servlet Capabilities, HTTP headers and MIME types Request Dispatcher: Including and forwarding, Sharing data with the request object attributes, Sharing data with Servlet Context attributes, Error Handling

**Unit-4:**

**A. RMI**: RMI Architecture, Designing RMI application, Executing RMI application, **B. Enterprise Java Beans:** Types of Enterprise Java beans, Session Bean & Entity Bean, Features of Session Bean, Life-cycle of Stateful, Seession Bean, Features of Entity Bean, Life-cycle of Entity Bean Container-managed Transactions & Bean-managed Transactions Implementing a container-managed Entity Bean

**Unit-5:**

**Java Struts and Hibernate**: Introduction to the Apache Struts, MVC Architecture, Struts Architecture, Struts Working, Introduction to the Struts Controller, Introduction to the Struts Action Class, Using Struts Action From Class, Using Struts HTML Tags, Introduction to Struts Validator Framework, Client Side Address Validation in Struts, Custom Validators Example Developing Application with Struts Tiles. Introduction to Hibernate, Hibernate framework 3.0, Hibernate Architecture, First Hibernate Application

**Reference Books**:

2. Java 2 Complete Reference by Herbert Schieldt (Sixth Edition)
3. Core Java Vol 1: Sun Press, Eighth Edition
4. Core Java Vol 2: Sun Press

**E-books**

- Java 2 Complete Reference by Herbert Schieldt (Fourth Edition)

**Lab Exercise for CSC458, Practical based on CSC411:**There should be minimum 20 lab assignment on the topics discussed in the course using Open Source Platform (Eclipse, Net Bean etc.).

## 6 Computer System Architecture

| | | | |
|---|---|---|---|
| **Course Code** | **CSC412** | **Course Title** | **Computer System Architecture** |
| **Number of Credits** | **3 Credits** (TH), **2 Credits** (PR) | **Internal** | **20%** |
| **Total Contact Hours** | **3 HRS** (TH/Week) **4 HRS** (PR/Week) | **External** (Semester/Term Exam) | **80%** |

**Course Outcomes**:

After complication of this course students will be able to understand and able to design circuit and simplify it. Will be able to understand in detail the how logic systems are built. Students will have thorough knowledge about

- Basic structure of a digital computer.
- Arithmetic operations of binary number system.
- The organization of the Control unit, Arithmetic and Logical unit, Memory unit and the I/O unit

**Course Outline:**

**Unit-1:**

**Data Representation:** Introduction, Data types, Complements, Fixed Point and Floating Point representation, Error Detecting Codes. **Simplification**

**Unit-2:**

**Combinational Circuits, Flip-Flops Sequential Circuits. Digital Logic Circuits:** Digital Computers, Logic Gates, Boolean algebra, Map

**Unit-3:**

**Digital Components:** Integrated Circuits, Decoders, Multiplexers, Registers, Shift Registers, Binary Counters, Memory Unit.

**Unit-4:**

**Central Processing Unit:** Introduction, general Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer (RISC). Serial **of Transfer, Direct Memory Access, Input-Output Processor**

**Unit-5:**

**I/O& Memory Organization:** Peripheral Devices, Input Output Interface, Asynchronous Data Transfer, Modes Communication, Memory, Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory, Memory Management Hardware.

**E-Books:**

http://www.tutorialspoint.com/computer_logical_organization/

**Lab Exercise:**

**CSC459 Practical based on CSC412:** At least one experiment should be carried out on each unit.

| Sr.No. | Title |
|---|---|
| 1. | Write a small study on Logisim software. |
| 2. | Describe the Python language with detail. |
| 3. | Describe math library of Python with its advantages. |
| 4. | Conversion of any base to decimal and decimal to any base. |
| 5. | Convert decimal value 1 to 20 into binary and also find out 1's and 2's compliment. |
| 6. | Implement logic diagram of tree variables EX-OR function with truth table of given expression x=A B C DeMorgan s theorem |
| 7. | Implement logic diagram using ⊕ ⊕ , (A+B)'+(A+B)'=0. |
| 8. | Implement logic diagram of given Boolean expression F=XYZ+XYZ+XYZ write its truth table. |
| 9. | Implement half adder and full adder with three input and write steps of its truth table. |
| 10. | Implement SR flip flop and write its truth table. |
| 11. | Implement four bit register and write its truth table. |
| 12. | Implement four bit shift register and write its truth table. |
| 13. | Implement 2:4 line decoder with NAND gate and writes its truth table. |
| 14. | To implement 4:1 line multiplexer and write its truth table. |
| 15. | Draw logic diagram of 2:4 line multiplexer and write its truth table. |

16. Draw block diagram of RAM, ROM and explain the types of memory.

17. Develop mathematical functions(Addition,Subtraction,Multiplication,Division,Mod,Log) using Python.

18. Program for Decimal to Binary conversion.

19. Program for Decimal to Binary conversion.

20. Program for Decimal to Binary conversion.

21. Program for Decimal to Binary conversion.

22. Program for Decimal to Binary conversion.

23. Program for Decimal to Binary conversion.

24. Program for Decimal to Binary conversion.

25. Program for Decimal to Binary conversion.

26. Program for I] Binary to Decimal

II] Decimal to Binary

III] Octal to Decimal

IV] Exit

27. Draw and implement logic gates using Logisim I] AND gate

II] OR gate

III] NOT gate

IV] NAND gate

V] NOR gate

VI] EX-OR gate

VII] EX-NOR gate

VIII] Half Adder

IX] Full Adder

X] Multiplexer

28. Develop mathematical

functions(Addition,Subtraction,Multiplication,Division,Mod,Log) using Python.

29. Program for Decimal to Binary conversion.

30. Program for Decimal to Binary conversion.

Computer System Architecture Practical will be conducted in two Softwares: for Programming, Open Source platform Python will be used and for circuit design, open source platform Multisim or Logisim will be used. The details of these are as follows:

## Semester - III

## 1. Advance Computer Networks

| | | | |
|---|---|---|---|
| **Course Code** | **CSC501** | **Course Title** | **Advance Computer Networks** |
| **Number of Credits** | **3 Credits** (TH)<br>**2 Credits** (PR) | **Internal** | **20** |
| **Total Contact Hours** | **3 HRS** (TH/Week)<br>**4 HRS** (PR/Week) | **External** (Semester/Term Exam) | **80** |

### Course Outcomes:

- As outcome student be able to do jobs or self-sustainable start-ups in the prominent areas like: Network Engineer, Network Administrator, Security provider, Network Analyst, Information Systems Administrator, Network Technician, and few more are: Computer Programmer, Computer Engineer, Database Administrator, Web Master, Computer-Operations Researcher, Computer Repair Specialist, Systems Analyst etc.

**Course Outline**

**Unit-1:**

**Introduction:** Background and overview of the layered architecture: Layered communication architecture: layers, services, protocols, layer entities, service access points, modes of communication, etc. DLL Protocols, Frame Relay, X.25 protocol, IEEE Standards for LAN, error detection and correction at DLL.

**Unit-2:**

**Internetworking and Routing:** Advanced Routing algorithms, Advanced Network Congestion Control algorithms, Packet Switching,IP Addressing & DNS.

**Unit-3:**

**Overview on Wireless Networks and Mobile Networks:** LAN, PAN, Sensor Networks, Ad_hoc Networks, Mobile IP, Mobile TCP, VPN,MAC Protocols. Wireless Protocols: Data Transport and Sensor Data Dissemination, Group Communication: Multicast Routing and Transport, Multicast (cont.); Scalability and Robustness in Networks.

**Unit-4:**

**Process to Process delivery:** client/server paradigm, multiplexing and demultiplexing, connectionless versus connection oriented services, reliable versus unreliable **Network Security:** Cryptography Techniques, Algorithms: Secret key and Public key, DES, RSA. Digital Signature, Firewalls, Proxy server.

**Unit-5:**

**Special Topics:** current and emerging trends: Next Generation Networks, Data Centers, Cloud Computing, GSM, GIS, Queuing models, ASN(abstract syntax Notation),distributed Networks.

**Books:**

William Stallings, Wireless Communications & Networks, 2nd edition, Prentice-Hall Pearson, 2005

**E-books:**

- http://newwayofengineering.blogspot.in/2014/12/computer-networks-tanenbaum-5th-edition.html
- http://nptel.iitm.ac.in/courses/Webcoursecontents/IIT%20Kharagpur/Computer%20Networks/New_index1.html
- http://nptel.iitm.ac.in/video.php?subjectId=1061050813.
- http://nptel.iitm.ac.in/courses/IIT-MADRAS/Computer_Networks/index.php

**Lab Exercise:**

**CSC552 Practical based on CSC501:** At least two experiments should be carried out on each unit.

## 2. Computer Graphics

| | | | |
|---|---|---|---|
| **Course Code** | **CSC502** | **Course Title** | **Computer Graphics** |
| **Number of Credits** | **3 Credits** (TH)<br>**2 Credits** (PR) | **Internal** | **20** |
| **Total Contact Hours** | **3 HRS** (TH/Week)<br>**4 HRS** (PR/Week) | **External** (Semester/Term Exam) | **80** |

### Course Outcomes:

- To understand the various computer graphics hardware and display technologies.
- 2D and 3D viewing technologies
- Various 2D and 3D objects transformation techniques.
- This course is beneficial for UGC-NET/SET examination and aptitude test.

**Course Outline:**

**Unit-1:**

**Introduction to Graphics:** The origin of computer graphics, How the interactive graphics display works. Display types: Random Scan and Raster Scan, Definitions: Pixel, Resolution, Aspect Ratio, Active graphics, Passive graphics, Interactive, Non interactive graphics, Application of Computer Graphics.

**Unit-2:**

**Line Drawing Technique:** Co-ordinate Systems, Incremental method, The Simple DDA, The Symmetrical DDA, Brenham s Algorithm.

**Unit-3:**

**Transformations:** Transformation principles, Concatenations, Matrix Representation, Three Dimensional Transformations, Transformation in Viewing, The perspective Transformation.

**Unit-4:**

**Clipping and Windowing:** Cohen-Sutherland algorithm, Mid-point Subdivision, Polygon Clipping, Viewing Transformation, The Windowing Transformation, 3-D Clipping.

**Unit-5:**

**Raster Graphics and Solid Area Scan –Conversion:** Introduction, Scan Converting Line Drawings, Scan Converting Polygons, Coherence, (YX) Algorithm.

**Reference Books:**

1. William M. Newman and Robert F. Sproull, Principles of Interactive Computer Graphics(Second Edition), Tata-McGraw Hill Publication.

1. Rogers, Procedural Interactive Computer Graphics, McGraw Hill Book Company Ltd.
2. Mathematical Elements of Interactive Computer Graphics, McGraw Hill Book Company Ltd.

**Lab Exercise:**

**CSC553 Practical based on CSC502: Practical: (Lab Work)**

Have a basic understanding of the core concepts of computer graphics.
Be capable of using Anim8or to create interactive computer graphics.
Have made pictures with their computer.
Make different games and animations .

1. Program for Brenham's line drawing
2. Program for circle generation
1. Program for line clipping
2. Program for 2D transformations
3. Program for 3D transformation
4. Animation

A mini project in Animation

## *Semester – IV*

## 1. Pattern Recognition

| | | | |
|---|---|---|---|
| **Course Code** | **CSC503** | **Course Title** | **Pattern Recognition** |
| **Number of Credits** | **3 Credits** (TH)<br>**2 Credits** (PR) | **Internal** | **20** |
| **Total Contact Hours** | **3 HRS** (TH/Week)<br>**4 HRS** (PR/Week) | **External** (Semester/Term Exam) | **80** |

### Course Outcomes:

- Student will be able to work with some prominent areas like Data & Applied Scientist, Data Scientist, Machine Learning Scientist, Principal Research Scientist, Sr Algorithm Engineer, Image Analysis Scientist, Lead Business Data Analyst, and Quantitative Researcher.

**Course Outline**

**Unit-1: Introduction to Pattern Recognition, Bayesian decision theory:** Classifiers, Discriminant functions, Decision surfaces, Normal density and Discriminant functions, discrete features

**Unit-2: methods, Maximum-Maximum Likelihood and Bayesian Estimation:** Parameter estimation Likelihood estimation, Bayesian estimation, Bayesian Parameter Estimation, Gaussian Case, General Theory, Problem of Dimensionality, Accuracy, Dimension, and Training Sample Size, Computational Complexity and Overfitting.

**Unit-3:EM , Hidden Markov models for sequential pattern classification, First-Order** Markov Component Models, Analysis First and Order Discriminants Hidden Markov: Principal Models, Component Hidden Markov Analyis Model (PCA),Computation, Exectation Evaluation, Decoding and Learning. **Density estimation, Parzen-window method, Probabilistic Neural Networks**

**Unit4: K-Nearest Neighbour, Estimation and rules, Nearest Neighbour and Fuzzy Classification. Nonparametric**

**Unit-5: Linear Discriminant function based classifiers:** Perceptron, Linear Programming Algorithm, Support Vector Machines (SVM)

**Books Recommended:**

- R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification 2nd Edition , John Wiley, 2007
- Christopher M. Bishop, Neural Network for Pattern Recognition , Oxford Ohio Press.

**References:**

1. E. Gose, R. Johansonbargh, Pattern Recognition and Image Analysis , PHI
2.EthenAlpaydin,**Neural**Introducti**Network**nto-AMachine**Classroom**Learning**Approach**,
3. SatishKumar, McGraw Hill.
4. Dr. Rao&Rao,Neural Network & Fuzzy Logic
5. S.Theodoridis and K.Koutroumbas, Pattern Recognition , 4th Ed., Academic Press,
6. C.M.Bishop, Pattern Recognition and Machine Learning , Springer, 2006

**Unit-2:**
Data Mining Classification: Basic Concepts, Decision Trees, and Model EvaluationClassification: Definition, Classification Techniques, Tree Induction, Measures of Node Impurity, Practical Issues of Classification, ROC curve, Confidence Interval for Accuracy, Comparing Performance of Two Models, Comparing Performance of Two Algorithms.Data Mining Classification: Alternative TechniquesRule-Based Classifier, Rule Ordering Schemes, Building Classification Rules, Instance-Based Classifiers, Nearest Neighbor Classifiers, Bayes Classifier, Naive Bayes Classifier, Artificial Neural Networks (ANN), Support Vector Machines.

**Unit-3:**
Data Mining Association Analysis: Basic Concepts and AlgorithmsAssociation Rule Mining, Frequent Itemset Generation, Association Rule Discovery : Hash tree, Factors Affecting Complexity, Maximal Frequent Horible Closed Item set, Alternative Methods for Frequent Item set Generation, FP-growth Algorithm, Tree Projection, Rule Generation, Pattern Evaluation, Statistical Independence, Properties of A Good Measure, Support-based Pruning, Subjective Interestingness Measure.

**Unit-4:**
Data Mining Cluster Analysis: Basic Concepts and AlgorithmsApplications of Cluster Analysis, Types of Clusters, Clustering Algorithms:K-means and its variants, Hierarchical clustering, Density-based clustering. Graph-Based Clustering, Limitations of Current Merging Schemes, Characteristics of Spatial Data Sets, Shared Near Neighbor Approach, ROCK (Robust Clustering using links), Jarvis-Patrick Clustering, SNN Clustering Algorithm.Data Mining Anomaly DetectionAnomaly/Outlier Detection, Importance, Anomaly Detection Schemes, Density-based: LOF approach.

**Unit-5:**
WEKA (Waikato Environment for Knowledge Analysis):is a well-known suite of machine learning software that supports several typical data mining tasks, particularly data preprocessing, clustering, classification, regression, visualization, and feature selection.Orange: is a component-based data mining and machine learning software suite that features friendly yet powerful, fast and versatile visual programming front-end for explorative data analysis and visualization, and Python bindings and libraries for scripting. It contains complete set of components for data preprocessing, feature scoring and filtering, modeling, model evaluation, and exploration techniques.RapidMiner:Formerly called YALE (Yet another Learning Environment), is an environment for machine learning and data mining experiments that is utilized for both research and real-world data mining tasks. HepWork:Designed for scientists, engineers and students, intelligible and comprehensive open-source data integration, processing, analysis, and exploration platform. It gives users the ability to visually create data flows or pipelines, selectively execute some or all analysis steps, and later study the results, models, and interactive views.

**REFERENCES**:

1.Introduction to Data Mining by Tan, Steinbach, Kumar.
2.Data Mining: Concepts and Techniques by Jiawei Han, Micheline Kamber, Morgan Kaufmann Publishers.
3.Data Mining: Practical Machine Learning Tools and Techniques by Ian H. Witten and Eibe