

SCSA1101	PYTHON AND PROBLEM SOLVING TECHNIQUES	L	T	P	Credits	Total Marks
		3	*	0	3	100

COURSE OBJECTIVES

- To learn fundamentals of PST & basics of python programming.
- To learn modular programming.
- To study in detailed data representation, operator, control structures, functions in python and apply it on problems.
- To learn in built functions in python & apply it on mathematical problem.
- To learn advanced algorithm by implementing lists, tuples and dictionaries.

UNIT 1 INTRODUCTION TO COMPUTER PROBLEM SOLVING**9 Hrs.**

Algorithms, building blocks of algorithms (statements, control flow, functions), notation (pseudo code, flow chart), algorithmic problem solving for socio economic conditions in global perspectives, simple strategies for developing algorithms (iteration, recursion), the efficiency of algorithms.

UNIT 2 DATA, EXPRESSIONS, STATEMENTS**9 Hrs.**

Python interpreter and interactive mode; values and types: int, float, boolean, string, and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; modules and functions, function definition and use, flow of execution, parameters and arguments; Algorithms: exchange the values of two variables, circulate the values of n variables, distance between two points.

UNIT 3 CONTROL FLOW, FUNCTIONS**9 Hrs.**

Conditional statements: Boolean values and operators, conditional (if), alternative (if-else), chained conditional (if-elif-else); Iterative statements: while, for, break, continue, pass, Applying for critical conditions. Algorithms: square root, exponentiation, sum an array of numbers, factorial computation-Sine function computation-Generation of the Fibonacci sequence-Reversing the digits of an integer.

UNIT 4 FACTORING METHOD**9 Hrs.**

Functions: return values, parameters, local and global scope. Lambda, filters, map and reduce functions. Algorithms: The smallest divisor of an integer-The greatest common divisor of two integers-Generating Prime Numbers-Computing the Prime Factors of an integer- Raising a Number to a Large Power

UNIT 5 LISTS, TUPLES, DICTIONARIES**9 Hrs.**

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing – list comprehension; Algorithms: Removal of Duplicates -Partitioning -Finding the kth smallest Element- histogram.

Max. 45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Design a modular solution for a given problem.
- CO2 - Choose appropriate algorithm for each module considering economic and social aspects.
- CO3 - Employ multiple critical and creative thinking strategies in reasoning and problem solving.
- CO4 - Evaluate and choose an appropriate problem solving strategy.
- CO5 - Document the approach to be implemented in any language.
- CO6 - Apply to the problem solving techniques on real world problems.

TEXT / REFERENCE BOOKS

1. Dromey.R.G, "How to Solve it by Computer", Prentice-Hall of India, 8th Indian Reprint, 2008.
2. Aho.A.V., Hopcroft.J.E and Ullman.J.D, "The Design and Analysis of Computer Algorithms", Pearson education, 2004.
3. Martin C. Brown, " Python: The Complete Reference", McGraw Hill, 2018.
4. John M. Zelle , "Python Programming: An Introduction to Computer Science", Library of Congress.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****PART A : 10 Questions of 2 marks each-No choice****PART B : 2 Questions from each unit with internal choice, each carrying 16 marks****Exam Duration : 3 Hrs.****20 Marks****80 Marks**

CSA1104	PROBLEM SOLVING TECHNIQUES WITH C AND C++	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To learn the fundamentals of PST and methodologies which are essential for building good C/C++ programs.
- To demonstrate a thorough understanding of modular programming by designing programs which require the use of programmer-defined functions.
- To impart the knowledge about pointers which is the backbone of effective memory handling.
- To demonstrate adeptness of object-oriented programming in developing solutions to problems demonstrating usage of data abstraction, encapsulation, and inheritance.

UNIT 1 INTRODUCTION TO COMPUTER PROBLEM SOLVING**9 Hrs.**

Algorithms - Building blocks of algorithms (statements, control flow, functions) -Notation (pseudo code, flow chart) - Algorithmic problem solving for socio economic conditions in global perspectives - Simple strategies for developing algorithms (iteration, recursion) - Efficiency of algorithms.

UNIT 2 BASICS OF C PROGRAMMING**9 Hrs.**

Introduction to C: Features of C - Structure of C program-Data Types-'C' Tokens-Input/output statements-Control Statement, Functions: – Types of Functions –Recursion.

Algorithms: Reversing the digits of a number - Generation of Fibonacci sequence- Factorial Computation.

UNIT 3 ARRAYS, STRINGS AND STRUCTURES**9 Hrs.**

Arrays: Single and Multidimensional Arrays— Array as Function Arguments, Strings: String Handling Functions, Structure: Nested Structures – Array of Structures – Structure as Function Argument–Function that Returns Structure, Union.

Algorithms: Sum of array elements- Removal of duplicates from an array-Finding the Kth smallest element.

UNIT 4 POINTERS AND FILE PROCESSING**9 Hrs.**

Pointers: Introduction, Arrays Using Pointers – Structures Using Pointers – Functions Using Pointer, Dynamic Memory Allocation, Storage Classes, File Handling in 'C'.

Algorithms: Swap elements using Call by Reference – Sorting Arrays using pointers- Finding sum of array elements using Dynamic Memory Allocation.

UNIT 5 OBJECT ORIENTED PROGRAMMING CONCEPTS**9 Hrs.**

Introduction-Procedure vs. object-oriented programming-Concepts: Classes and Objects-Operator & Function Overloading-Inheritance-Polymorphism and Virtual Functions.

Max.45 Hrs.**COURSE OUTCOMES**

On completion of the course the student will be able to

- CO1:** Develop solutions for the given problem.
- CO2:** Design and Implement applications using arrays and strings.
- CO3:** Develop and Implement applications using memory allocation and File concepts.
- CO4:** Use proper class protection to provide security.
- CO5:** Describe the reusability of code through Inheritance.
- CO6:** Demonstrate the use of virtual functions to implement polymorphism.

TEXT / REFERENCE BOOKS

1. Dromey.R.G, "How to Solve it by Computer", Prentice-Hall of India, 8th Indian Reprint, 2008.
2. Aho.A.V.,Hopcroft.J.E and Ullman.J.D, "The Design and Analysis of Computer Algorithms", Pearson education, 2004.Deitel and Deitel, C how to Program, 7th Edition, Pearson Education, 2013.
3. Brian W.Kernighan and Dennis M.Ritchie, The C Programming Language, Pearson Education, 2015.
4. Bhave, Object Oriented Programming with C++, Pearson Education, 2004.
5. John R Hubbard, "Programming with C++", Schaums Outline Series, McGraw Hill, 2nd edition, 2009.
6. Bjarne Stroustrup, Programming: Principles and Practice using C++, 1st Edition, Addison Wesley Publications, 2008.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A :** 10 Questions of 2 marks each-No choice**20 Marks****PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**80 Marks**

SCSA1201	FUNDAMENTALS OF DIGITAL SYSTEMS	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To understand number systems and codes.
- To illustrate simplified Boolean expressions using Gates.
- To construct combinational logic circuits.
- To design sequential logic circuits.
- To analyze circuits and latches.

UNIT 1 NUMBER SYSTEMS, COMPLIMENTS AND CODES**9 Hrs.**

Number Systems – Binary Numbers-Number base conversions-Octal and Hexa Decimal Numbers – Complements –Signed Binary Numbers-Binary Arithmetic –Binary Codes-Decimal Code-Error Detection code-Gray Code- Reflection and Self Complementary codes-BCD number representation – Alphanumeric codes ASCII/EBCDIC –Hamming Code- Generation, Error Correction.

UNIT 2 BOOLEAN ALGEBRA AND LOGIC GATES**9 Hrs.**

Axiomatic definitions of Boolean Algebra – Basic Theorems and Properties of Boolean Algebra – Boolean Functions-Canonical and Standard forms-Digital Logic Gates– Simplification of Boolean Expressions: The map method- SOP and POS – NAND and NOR implementation-Don't Cares –The Tabulation Method-Determination and Selection of Prime Implicants.

UNIT 3 COMBINATIONAL LOGIC**9 Hrs.**

Design Procedure-Adder – Subtractor – Code Conversion – Analysis Procedure –Multilevel NAND/NOR circuits-Exclusive OR functions – Binary adder and subtractor– Decimal adder – BCD adder – Magnitude Comparator – Decoders – Demultiplexer – Encoder – Multiplexers.

UNIT 4 SYNCHRONOUS SEQUENTIAL LOGIC**9 Hrs.**

Flip Flops – Analysis of clocked sequential circuit –Reduction and Assignments–Flip flop excitation tables-Design Procedure-Design of counters-Registers-Shift registers-Synchronous Counters-Timing sequences-Algorithmic State Machines-ASM chart-timing considerations-control implementation.

UNIT 5 ASYNCHRONOUS SEQUENTIAL LOGIC AND MEMORY UNIT**9 Hrs.**

Circuits with Latches-Analysis procedure and Design Procedure-Reduction of state and Flow tables-Race –Free State Assignment.

Max. 45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Perform conversions between number systems.
- CO2 - Simplify Boolean expressions and model using gates.
- CO3 - Discover the principles behind combinational logic circuits used in real time.
- CO4 - Survey the flip-flops needed for sequential logic circuits design.
- CO5 - Analyze the sequential logic circuits.
- CO6 - Discuss about memory unit and arithmetic logic unit.

TEXT / REFERENCE BOOKS

1. Morris Mano, "Digital Logic and Computer Design", Prentice Hall India, 2006.
2. Thomas L Floyd, "Digital Fundamentals", 10th Edition, Pearson Education, 2009.
3. A.P.Malvino and D.P.Leach, "Digital Principles and Applications", 6th Edition, McGraw-Hill, 2006.
4. Thomas C. Bartee, "Computer Architecture Logic Design", 3rd Edition, 2002.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A :** 10 Questions of 2 marks each-No choice**20 Marks****PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**80 Marks**

SCSA1203	DATA STRUCTURES	L	T	P	Credits	Total Marks
		3	*	0	3	100

COURSE OBJECTIVES

- To impart the basic concepts of data structures and algorithms.
- To be familiar with writing recursive methods.
- To understand concepts about searching and sorting techniques.
- To implement basic concepts about stacks.
- To apply the concepts of queues and its types.

UNIT 1 INTRODUCTION TO ALGORITHMS**9 Hrs.**

Introduction Data Structures - Need - classification - operations –Abstract data types (ADT) - Array - characteristics - types - storage representations. Array Order Reversal-Array Counting or Histogram-Finding the maximum Number in a Set, Recursion- Towers of Hanoi-Fibonacci series-Factorial.

UNIT 2 LINKED LISTS**9 Hrs.**

Introduction - Singly linked list - Representation of a linked list in memory - Operations on a singly linked list - Merging two singly linked lists into one list - Reversing a singly linked list - Applications of singly linked list to represent polynomial - Advantages and disadvantages of singly linked list - Circular linked list - Doubly linked list - Circular Doubly Linked List.

UNIT 3 STACKS**9 Hrs.**

Basic Stack Operations - Representation of a Stack using Arrays - Algorithm for Stack Operations - Stack Applications: Reversing list - Factorial Calculation - Infix to postfix Transformation - Evaluating Arithmetic Expressions.

UNIT 4 QUEUES**9 Hrs.**

Basic Queue Operations - Representation of a Queue using array - Applications of Queues - Round robin Algorithm - Enqueue - Dequeue - Circular Queues - Priority Queues.

UNIT 5 SEARCHING AND SORTING TECHNIQUES**9 Hrs.**

Basic concepts - List Searches using Linear Search - Binary Search - Fibonacci Search - Sorting Techniques - Insertion sort - Heap sort - Bubble sort - Quick sort - Merge sort - Analysis of sorting techniques.

Max. 45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Understand the concept of recursive algorithms.
- CO2 - Demonstrate the different types of data structures.
- CO3 - Able to understand the operations on linear data structures.
- CO4 - Summarize searching and sorting techniques.
- CO5 - Choose appropriate data structure as applied to specified problem definition.
- CO6 - Understand and implement the applications of linear data structures.

TEXT / REFERENCE BOOKS

1. Jean-Paul Tremblay, Paul G. Sorenson, 'An Introduction to Data Structures with Application', TMH, 2017.
2. Richard F. Gilberg, Forouzan, "Data Structures", Cengage, 2004, 2nd Edition.
3. Larry R. Nyhoff, ADTs, Data Structures, and Problem Solving with C++, Prentice Hall Editin, 2004.
4. Thomas H. Cormen, Charles E. Leiserson, "Introduction to Algorithms", 3rd Edition, 2010.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A** : 10 Questions of 2 marks each-No choice**20 Marks****PART B** : 2 Questions from each unit with internal choice, each carrying 16 marks**80 Marks**

SCSA1204	PYTHON PROGRAMMING	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To understand why Python is a useful scripting language for developers.
- To learn how to use lists, tuples, and dictionaries in Python programs.
- To build and package Python modules for reusability.
- To understand how to read and write files in Python.
- To learn how to use exception handling in Python applications for error handling.
- To design and program Python applications.

UNIT 1 INTRODUCTION**9Hrs.**

History of Python- Introduction to the IDLE interpreter (shell) - Data Types - Built-in function - Conditional statements - Iterative statements- Input/output functions - Compound Data Types - Nested compound statements – Introduction to Object Oriented Concepts.

UNIT 2 FILES AND EXCEPTIONS HANDLING, MODULES, PACKAGES**9Hrs.**

File Operations –Iterators - Exception handling - Regular Expressions- Creating Modules-Import Statement- Introduction to PIP-Installing Packages via PIP-Using Python Packages.

UNIT 3 GUI PROGRAMMING**9 Hrs.**

GUI Programming in Python - Introduction to GUI library - Layout management - Events and bindings - Fonts – Colours - Canvas - Widgets (frame, label, button, check box, entry, listbox, message, radiobutton, text, spinbox).

UNIT 4 DATABASE AND NETWORK**9Hrs.**

Database (using NoSQL): Connector Module –Cursor – Statements - Exceptions in database. Network connectivity: Socket module - Client – Server – Email – URLAccess.

UNIT 5 CASE STUDY**9Hrs.**

Web Programming using Python Image Processing – Facebook Analysis – Twitter Analysis.

Max. 45 Hrs.**COURSE OUTCOMES**

- CO1:** Describe the Numbers, Math functions, Strings, List, Tuples and Dictionaries in Python.
- CO2:** Do the decision making and write functions in Python.
- CO3:** Explain how to design GUI Applications in Python and evaluate different database operations.
- CO4:** Design and develop Client Server network applications using Python.
- CO5:** Ability to design real life situational problems and think creatively about solutions of them.
- CO6:** Apply the best features of mathematics, engineering and natural sciences to program real life problems.

TEXT /REFERENCE BOOKS

1. Y. Daniel Liang, "Introduction to Programming Using Python", Pearson, 2013.
2. Python Notes for Professionals by Stack Overflow Documentation (<https://books.goalkicker.com/PythonBook/>)
3. Dr. Charles R. Severance, "Python for Everybody- Exploring Data Using Python 3", 2016.
4. Paul Gries, Jennifer Campbell, Jason Montojo, "Practical Programming: An Introduction to Computer Science using Python 3", Pragmatic Bookshelf, 2nd Edition, 2014.
5. Magnus Lie Hetland, "Beginning Python: From Novice to Professional", Apress.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A :** 10 Questions of 2 marks each-No choice**20 Marks****PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**80 Marks**

SCSA1301	DATABASE MANAGEMENT SYSTEMS	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To understand terms related to database design and management.
- To gain knowledge in relational model and relational database management system.
- To implement relational databases using SQL & My SQL.
- To understand database security and performance issues.
- To understand the basics of Data warehousing and Data mining.

UNIT 1 INTRODUCTION TO DATABASES**9 Hrs.**

Databases and database users – Database system concepts and architecture – Data modeling using entity Relationship(ER) model – Enhanced ER model- Relational Model - The Relational Data Model and Relational Database Constraints - The Relational Algebra and Relational Calculus.

UNIT 2 DATABASE DESIGN**9 Hrs.**

Overview of the Hierarchical Data Model - Overview of the Network Data Model – Relational database design: Mapping ER Model to Relational Model - Commercial query languages: QBE – Functional dependency – Normalization.

UNIT 3 QUERY PROCESSING**9 Hrs.**

SQL Queries -Embedded SQL -My SQL: Basics, Queries in MySQL and Algorithms for Query Processing and Optimization - Introduction to Transaction Processing Concepts and Theory - Concurrency control techniques.

UNIT 4 RECOVERY AND SECURITY**9 Hrs.**

Database Recovery Techniques - Database Security – Debate on the distributed databases and Client- Server Architecture with reference to Indian Railway Reservation System.

UNIT 5 OBJECT DATABASE AND CURRENT TRENDS**9 Hrs.**

Concepts for Object Database - Emerging Database Technologies and Application - Introduction to Data warehousing & Data mining –Applications of Data mining.

Max. 45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Design ER-models to represent simple database application scenarios.
- CO2 - Ability to understand and design data modelling using Entity-Relationship model.
- CO3 - Implement SQL to a broad range of query and data update problems.
- CO4 - Articulate socio-economic applications of distributed databases and use database recovery mechanisms.
- CO5 - Familiar with data warehousing and data mining applications.
- CO6 - Apply Normalization techniques to normalize and improve the database design.

TEXT / REFERENCE BOOKS

1. Elmasri & Navathe, "Fundamentals of Database Systems", 6th Edition, Addison Wesley, 2011.
2. Abraham Silberschatz, Henry.F.Korth and S.Sudharshan, "Database System Concepts", 4th Edition, 2002.
3. Jiawei Han and Micheline Kamber, "Data Mining Concepts and Techniques", 3rd Edition, Elsevier, 2012.
4. Jan L. Harrington, "Object oriented database design", Harcourt India private limited 2000.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****PART A :** 10 Questions of 2 marks each-No choice**PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**Exam Duration : 3 Hrs.****20 Marks****80 Marks**

SCSA1302	THEORY OF COMPUTATION	L	T	P	Credits	Total Marks
		3	*	0	3	100

COURSE OBJECTIVES

- To introduce Automata Theory and Regular Languages.
- To understand Context Free languages and recognizers for different languages.
- To design Turing Machines for various languages.
- To understand the concepts involved in software development.
- To gain knowledge on undecidable problems.

UNIT 1 FINITE AUTOMATA AND REGULAR LANGUAGES 9 Hrs.

Finite automata and regular languages - Regular languages and regular expressions - Finite automata -Non-determinism and Kleene's theorem - Non-deterministic finite automata and NFA with null transition.

UNIT 2 CONTEXT-FREE LANGUAGES AND NORMAL FORMS 9 Hrs.

Context-free grammars - Definition - More examples - Union, concatenations, and *'s of CFLs - Derivation trees and ambiguity - Unambiguous CFG for algebraic expressions - Normal Forms - CNF – GNF.

UNIT 3 PUSH DOWN AUTOMATA 9 Hrs.

Pushdown automata - Introduction - Definition - Deterministic pushdown automata - PDA corresponding to a given context-free grammar – Context-free Grammar corresponding to PDA. Pumping Lemma for CFG.

UNIT 4 TURING MACHINES 9 Hrs.

Turing machines - Models of computation and the Turing thesis - Definition of TM and TM as language acceptor - Non-deterministic TM and Deterministic TM – Universal TM.

UNIT 5 RECURSIVE LANGUAGES AND UNDECIDABILITY 9 Hrs.

Recursively enumerable and recursive languages – Properties of Recursively enumerable and recursive languages - Enumerating a language. Introduction to Undecidability- Halting problem-Undecidability of Post correspondence problem (PCP)-Modified PCP -Rice Theorem.

Max. 45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Identify, analyze and compute a solution to a problem.
- CO2 - Interpret data using computational theory.
- CO3 - Determine whether a language is context free or not.
- CO4 - Design a Turing Machine for a given problem.
- CO5 - Apply the theoretical knowledge the design of compilers.
- CO6 - Identify the limitations of some computational models.

TEXT / REFERENCE BOOKS

1. Introduction to Languages and the Theory of Computation, John. C. Martin, Tata McGraw-Hill, 2003.
2. Introduction to Automata Theory, Languages and Computation, Hopcroft, Motwani, and Ullman, Pearson Publishers, 3rd Edition, 2006.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****PART A : 10 Questions of 2 marks each-No choice****PART B : 2 Questions from each unit with internal choice, each carrying 16 marks****Exam Duration : 3 Hrs.****20 Marks****80 Marks**

SCSA1303	SOFTWARE ENGINEERING	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- Software process models and compare their applicability.
- Identify the key activities in managing a software project.
- Concepts of requirements engineering and Analysis Modelling.
- Apply systematic procedure for software design and deployment.
- Compare and contrast the various testing and maintenance.

UNIT 1 SOFTWARE PROCESS AND PROJECT MANAGEMENT**9 Hrs.**

The evolving role of software – the changing nature of software- Life cycle models - Water fall - Incremental - Spiral - Evolutionary - Prototyping – Concurrent development – Specialised process models - Verification - Validation - Life cycle process - Development process - System engineering hierarchy - Introduction to CMM - Levels of CMM.

UNIT 2 REQUIREMENT ANALYSIS AND SPECIFICATIONS**9 Hrs.**

Functional And Non-Functional - User - System - Requirement Engineering Process - Feasibility Studies –communication practices- Requirements - Elicitation - Validation and management - Fundamental of requirement analysis – Analysis principles – Structured System Analysis - Software prototyping - Prototyping in the Software Process - Data - Functional and Behavioral Models - Structured Analysis and Data Dictionary.

UNIT 3 SOFTWARE DESIGN**9 Hrs.**

Design process - Modular design - Design heuristic - Design model and document - Architectural design - Software architecture - Data design - Architecture data - Transform and transaction mapping - User interface design - User interface design principles.

UNIT 4 TESTING AND IMPLEMENTATION**9 Hrs.**

Levels - Software Testing Fundamentals - Types of s/w test - White box testing- Basis path testing - Black box testing - Control Structure testing- Regression testing strategies - Strategic approach and issues - UNIT testing - Integration testing - Validation testing - System testing and debugging. Case studies - Writing black box and white box testing-Coding Practices-Refactoring.

UNIT 5 PROJECT MANAGEMENT AND ESTIMATION**9 Hrs.**

Software cost estimation - COCOMO model - Quality management - Quality concepts- SQA - Software reviews - Formal technical reviews - Formal approaches of SQA and software reliability - Software maintenance - SCM - Need for SCM - Version control - Introduction to SCM process - Software configuration items. Re-Engineering - Software reengineering - Reverse engineering - Restructuring - Forward engineering.

Max. 45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Identify and apply software lifecycle model for a given problem and will know the criteria for each level of CMM.
- CO2 - Comprehend types of requirements and summarize Requirement Engineering Process.
- CO3 - Design data, functional and behavioral model for any given software requirement.
- CO4 - Identify and analyze levels of testing and perform white box testing and black box testing for a given problem.
- CO5 - Describe concepts of software quality assurance and software configuration management.
- CO6 - Compare and contrast forward engineering, reverse engineering and reengineering.

TEXT / REFERENCE BOOKS

1. Pressman, "Software Engineering and Application", 7th Edition, McGraw International Edition, 2009.
2. Ian Sommerville, "Software Engineering", 8th Edition, Pearson Education, 2008.
3. Stephan Schach, "Software Engineering", Tata McGraw Hill, 2007.
4. Pfleeger and Lawrance, "Software Engineering: Theory and Practice" Pearson Education, 2nd Edition, 200.1
5. Rajib Mall, "Fundamentals of Software Engineering", 3rd Edition, PHI Learning Private Limited, 2009.
6. Pankaj Jalote, "Software Engineering, A Precise Approach", Wiley India, 2010.
7. Kelkar S.A., "Software Engineering", Prentice Hall of India Pvt Ltd, 2007.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****PART A :** 10 Questions of 2 marks each-No choice**PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**Exam Duration : 3 Hrs.****20 Marks****80 Marks**

SCSA1304	ADVANCED DATA STRUCTURES	L	T	P	Credits	Total Marks
		3	*	0	3	100

COURSE OBJECTIVES

- To acquire knowledge of organizing the data in non linear fashion.
- To get the idea of balancing the height of trees to optimize the structure and search time.
- To learn the process of establishing the network with various nodes with minimum cost and finding the shortest path.
- To understand the method of designing the table data structure and its applications.

UNIT 1 BASIC TREE CONCEPTS**9 Hrs.**

Trees- Ordinary and Binary trees terminology, Properties of Binary trees, Implementation using Array and Linked list - Binary tree ADT representations, recursive and non recursive traversals - Binary Search Tree - Insertion and Deletion.

UNIT 2 ADVANCED TREE CONCEPTS**9 Hrs.**

Threaded Binary Trees, AVL Tree, B-tree Insertion and deletion, Splay trees - Heap trees - Heapify Procedure, Tries.

UNIT 3 GRAPH CONCEPTS**9 Hrs.**

Terminology, Representation using Array and Linked List - Types of graphs - Graph traversals - BFS and DFS - Applications.

UNIT 4 ADVANCED GRAPH CONCEPTS**9 Hrs.**

Minimum Spanning Tree - Kruskal's, Prim's and Sollin's Algorithm - Shortest path using Dijkstra's, Bellman Ford and Floyd Warshall Algorithm.

UNIT 5 TABLES AND SETS**9 Hrs.**

Rectangular tables - Jagged tables - Inverted tables - Symbol tables - Static tree tables - Dynamic tree tables - Hash tables. Sets: Representation - Operations on sets - Applications.

Max. 45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Design and implement the various non-linear data structures and perform the intended operations.
- CO2 - Understand the strength of balancing the height of the trees.
- CO3 - Analyze the time complexity of various non linear data structures.
- CO4 - Apply the algorithms to find the shortest path & to connect the nodes with minimum cost.
- CO5 - Design the table and applying the table for many applications.
- CO6 - Compare and identify the usage of different data structures.

TEXT / REFERENCE BOOKS

1. Ellis Horowitz and Sartaj Sahni "Fundamentals of Data Structures" Galgotia Book Source, Pvt. Ltd., 2004.
2. M. A. Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition, Pearson Education, 2005.
3. Jean Paul Tremblay and Paul G. Sorenson, "An Introduction to Data Structures with Applications", Tata McGraw-Hill, 2nd edition, 2001.
4. Aaron M Tanenbaum, Moshe J Augenstein and Yedidyah Langsam, "Data Structures using C and C++", Pearson Education, 2004.
5. V. Aho, J.E. Hopcroft, and J.D. Ullman, "Data Structures and Algorithms", Pearson Education, 1st Edition Reprint 2003.
6. R. F. Gilberg, B. A. Forouzan, "Data Structures", 2nd Edition, Thomson India Edition, 2005.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks: 100****Exam Duration: 3 Hrs.****PART A :** 10 Questions of 2 marks each-No choice**20 Marks****PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**80 Marks**

SCSA1306	DESIGN THINKING	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To understand the principles of Design Thinking, a creative solution-based approach to problem solving.
- To understand about Agile methodology as a practice to promote continuous iteration of development and testing throughout SDLC.
- To understand the basics about development cycles, IT Operations & faster innovation.
- To understand the practice of design thinking for Strategic Innovation
- To understand DevOps the advanced process of software engineering for faster problem resolution & team collaboration.

UNIT 1 INTRODUCTION TO DESIGN THINKING**9 Hrs.**

Introduction to Design Thinking – Importance of Design Thinking – History of Design Thinking- Design Thinking Framework - Design Thinking Methods - Empathise –Define – Ideate – Prototype – Test- Software Development Methodology – Waterfall model – V –model -Customer Example.

UNIT 2 INTRODUCTION TO AGILE**9 Hrs.**

History of Agile – Agile principles – Agile Vs Waterfall – Agile Methodology Overview- Agile frameworks – Extreme programming - Rational Unified Process (RUP) - Test Driven Development (TDD) – Feature Drive Development (FDD)- Scrum - Kanban Methodology – Agile and Devops.

UNIT 3 AGILE SOFTWARE DEVELOPMENT**9 Hrs.**

Software Development- using Extreme Programming – Roles & Rules - Software Development using Scrum Framework – Scrum team – Sprints – Sprints planning – Metrics – Scrum tools - Case Studies.

Unit 4 DESIGN THINKING FOR STRATEGIC INNOVATION**9 Hrs.**

Innovation Management-Changing Management Paradigms-Design Thinking related to Science and art-Design Thinking in Business-Linking Design Thinking Solution to Business Challenges.

UNIT 5 DEVOPS**9 Hrs.**

Introduction to DevOps – DevOpsvs Agile – DevOps Principles and Life Cycle – Introduction to CI / CD &DevOps Tools–Version Control – Build Automation – Configuration Management – Containerization – Continuous Deployment – Continuous Integration – Continuous Testing –Continuous Monitoring.

Max.45 Hours**COURSE OUTCOMES**

On completion of the course the student will be able to

- CO1:** Apply design thinking concepts to give solution for the problems identified
- CO2:** Implement Agile software methodology for faster development of quality software
- CO3:** Describe how to improve collaboration between development and operations.
- CO4:** Design innovative products
- CO5:** Implement Automated Installations and Deployments
- CO6:** Unresolve different transformations of a product or a service through brainstorming and incremental approach, etc.

TEXT /REFERENCE BOOKS

1. MauricioVianna, YsmarVianna, Brenda Lucena and Beatriz Russo," Design thinking : Business innovation", MJV Technologies and innovation press, 2011.
2. Design Thinking: Integrating Innovation, Customer Experience, and Brand Valueby Thomas Lockwood (Editor) Published February 16th 2010 by Allworth Press.
3. KalloriVikram, —Introduction to DevOps, 1 st Edition, KalloriVikram Publication, 2016.
4. Jaokim Verona, —Practical DevOps, 2 nd Edition, Packt. Publication, 2018.
5. Stephen Fleming, Pravin, —DevOps Handbook: Introduction of DevOps Resource Management—, 1st Edition, Createspace Independent Pub. , 2010.

6. Len Bass, Ingo Weber, Liming Zhu, G., —DevOps: A Software Architect's Perspective, 1st Edition, Addison-Wesley Professional, 2015.
7. Alistair Cockburn, "Agile Software Development", 2nd ed, Pearson Education, 2007.

END SEMESTER EXAM QUESTION PAPER PATTERN

Max. Marks : 100

Exam Duration : 3 Hrs.

PART A : 10 Questions of 2 marks each-No choice

20 Marks

PART B : 2 Questions from each unit with internal choice, each carrying 16 marks

80 Marks

SCSA1401	OBJECT ORIENTED ANALYSIS AND SYSTEM ENGINEERING	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To understand the fundamentals of Object Oriented System Development.
- To understand the object oriented methodologies.
- To use UML in requirements elicitation and designing.
- To understand concepts of relationships and aggregations.
- To test the software against its requirements specification.

UNIT 1 AN OVERVIEW OF OBJECT ORIENTED SYSTEM DEVELOPMENT**9 Hrs.**

Introduction - Object Oriented System Development Methodology - Why Object Orientation - Overview of Unified Approach - Object Basics: Object Oriented Philosophy - Objects - Classes - Attributes - Object Behavior and Methods, Messages and Interfaces, Encapsulation and Information Hiding - Class Hierarchy - Polymorphism - Object Relationships and Associations - Aggregations and Object Containment - Object Identity - Static and Dynamic Binding - Persistence. Object-oriented CASE tools, Object Oriented Systems Development Life Cycle: Software Development Process - Building High Quality Software - Use case Driven Approach – Reusability.

UNIT 2 OBJECT ORIENTED METHODOLOGIES**9 Hrs.**

Rumbaugh et al.'s Object Modeling Technique - Booch Methodology - Jacobson et al. Methodologies – Patterns - Framework - Unified approach - Unified Modeling Language: Static and Dynamic Model - UML Diagrams - UML Class Diagram – UML Use Case –Case study- Use case Modelling – Relating Use cases – include, extend and generalization – When to use Use-cases- UML Dynamic Modeling –Case study- UML Extensibility - UML Metamodel.

UNIT 3 OBJECT ORIENTED ANALYSIS**9 Hrs.**

Business Object Analysis - Use Case Driven Object Oriented Analysis - Business Process Modeling - Use Case model - Developing Effective Documentation - Object Analysis Classification: Classification Theory - Noun Phrase Approach - Common Class Patterns Approach - **Use-Case Driven Approach** - Classes Responsibilities and Collaborators - Naming Classes - Identifying Object Relationships, Attributes and Methods: Association – SuperSubclass Relationships - A-part of Relationships.

UNIT 4 OBJECT ORIENTED DESIGN**9 Hrs.**

Object Oriented Design Process - **Object Oriented Design Axioms** - Corollaries - Designing Classes: Object Constraint Language - Process of Designing Class - Class Visibility - Refining Attributes - Access Layer: Object Store and Persistence - Database Management System - **Logical and Physical Database Organization and Access Control** - Distributed Databases and Client Server Computing - Object Oriented Database Management System – Object Relational Systems - Designing Access Layer Classes - **View Layer: Designing View Layer Classes** - **Macro Level Process** - **Micro Level Process** - **Purpose of View Layer Interface** - Prototyping the user interface.

UNIT 5 SOFTWARE QUALITY**9 Hrs.**

Software Quality Assurance- **Impact of Object Orientation on Testing** - **Develop Test Cases and Test Plans** - **System Usability and Measuring User Satisfaction: Usability Testing** - **User Satisfaction Testing**.

Max. 45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Understand the basics object model for System development.
- CO2 - Understand the object Oriented Methodologies.
- CO3 - Express software design with UML diagrams.
- CO4 - Understand the concept of Relationships.
- CO5 - Design software applications using OO concepts.
- CO6 - Understand the various testing methodologies for OO software.

TEXT / REFERENCE BOOKS

1. Ali Bahrami, "Object oriented systems development using the unified modelling language", McGraw- Hill.
2. Grady Booch, James Rumbaugh, and Ivar Jacobson, "The Unified Modeling Language User Guide", 3rd Edition Addison Wesley.
3. John Deacon, "Object Oriented Analysis and Design", 1st Edition, Addison Wesley.
4. Bernd Oestereich, "Developing Software with UML, Object - Oriented Analysis and Design in Practice", Addison-Wesley.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks: 100****Exam Duration: 3 Hrs.****PART A:** 10 Questions of 2 marks each-No choice**20 Marks****PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**80 Marks**

SCSA1402	COMPUTER ARCHITECTURE AND ORGANIZATION	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To impart knowledge on various types of Registers and Microinstructions.
- To discuss about the Data path design and Processor design in detail.
- To understand the types of memory organizations.
- To discuss about the registers, interface and communication in I/O devices.
- To analyze about the characteristics, structure, communication and synchronization of multiprocessors.

UNIT 1 CENTRAL PROCESSING UNIT**9 Hrs.**

Introduction - General Register Organization - Stack organization - Basic computer Organization - Instruction codes - Computer Registers - Computer Instructions - Instruction Cycle - Arithmetic – Logic - Shift Micro operations - Arithmetic Logic Shift unit - Example Architectures: MIPS – Power – PC – RISC – CISC.

UNIT 2 COMPUTER ARITHMETIC**9 Hrs.**

Addition - Subtraction - Multiplication and Division algorithms - Floating Point Arithmetic operations Micro programmed Control - Control memory - address sequencing – Micro program Example - Design of Control unit - Example Processor design.

UNIT 3 MEMORY ORGANIZATION**9 Hrs.**

Memory Hierarchy - Main memory - auxiliary Memory - Associative Memory - Cache Memory - Virtual memory.

UNIT 4 INPUT - OUTPUT ORGANIZATION**9 Hrs.**

Peripheral Devices - I/O Interface - Modes of transfer - Priority Interrupt - DMA - IOP - Serial Communication.

UNIT 5 CHARACTERISTICS OF MULTIPROCESSORS**9 Hrs.**

Interconnection Structures - Interprocessor Arbitration - Interprocessor Communication and Synchronization - Cache coherence.

Max. 45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Classify the various types of registers, microinstructions and addressing modes.
- CO2 - Design the data path for processor.
- CO3 - Demonstrate the usage of Memory Organization.
- CO4 - Describe about the I/O devices.
- CO5 - Explain the characteristics, structure, communication and synchronization of multiprocessor.
- CO6 - State the inter processor communication and synchronization.

TEXT / REFERENCE BOOKS

1. M.Morris Mano, "Computer system Architecture", 3rd Edition, Prentice-Hall Publishers, 2007.
2. Mark Burrell, "Fundamentals of Computer Architecture", Mcmillan Higher Education, 2003.
3. John D. Carpinelli, "Computer Systems Organization and Architecture", Pearson Education, 2001.
4. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", 5th Edition, McGraw-Hill, 2002.
5. William Stallings, "Computer Organization and Architecture - Designing for Performance", 9th Edition, Prentice Hall, 2012.
6. John P Hayes, Computer Architecture Organization, McGraw Hill Edition 4, 2003.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****PART A :** 10 Questions of 2 marks each-No choice**PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**Exam Duration : 3 Hrs.****20 Marks****80 Marks**

SCSA1403	DESIGN AND ANALYSIS OF ALGORITHMS	L	T	P	Credits	Total Marks
		3	*	0	3	100

COURSE OBJECTIVES

- To analyze the performance of algorithms under various scenarios.
- To learn mathematical background for algorithm analysis & solving the recurrence equations.
- To learn various algorithm design techniques.
- To understand and apply the algorithms.

UNIT 1 INTRODUCTION**9 Hrs.**

Fundamentals of Algorithmic Problem Solving - Time Complexity - Space complexity with examples - Growth of Functions - Asymptotic Notations: Need, Types - Big Oh, Little Oh, Omega, Theta - Properties - Complexity Analysis Examples - Performance measurement - Instance Size, Test Data, Experimental setup.

UNIT 2 MATHEMATICAL FOUNDATIONS**9 Hrs.**

Solving Recurrence Equations - Substitution Method - Recursion Tree Method - Master Method - Best Case - Worst Case - Average Case Analysis - Sorting in Linear Time - Lower bounds for Sorting: - Counting Sort - Radix Sort - Bucket Sort.

UNIT 3 BRUTE FORCE AND DIVIDE-AND-CONQUER**9 Hrs.**

Brute Force:- Travelling Salesman Problem - Knapsack Problem - Assignment Problem - Closest Pair and Convex Hull Problems - Divide and Conquer Approach:- Binary Search - Quick Sort - Merge Sort - Strassen's Matrix Multiplication.

UNIT 4 GREEDY APPROACH AND DYNAMIC PROGRAMMING**9 Hrs.**

Greedy Approach:- Optimal Merge Patterns- Huffman Code - Job Sequencing problem- -- Tree Vertex Splitting Dynamic Programming:- Dice Throw-- Optimal Binary Search Algorithms.

UNIT 5 BACKTRACKING AND BRANCH AND BOUND**9 Hrs.**

Backtracking:- 8 Queens - Hamiltonian Circuit Problem - Branch and Bound - Assignment Problem - Knapsack Problem:- Travelling Salesman Problem - NP Complete Problems - Clique Problem - Vertex Cover Problem .

Max. 45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Determine the suitable algorithmic design technique for a given problem.
- CO2 - Identify the limitations of algorithms in problem solving.
- CO3 - Analyze the efficiency of the algorithm based on time and space complexity.
- CO4 - Implement asymptotic notations to analyze worst-case and average case running times of algorithms.
- CO5 - Interpret the fundamental needs of algorithms in problem solving.
- CO6 - Describe the various algorithmic techniques and its real time applications.

TEXT/REFERENCE BOOKS

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", 3rd Edition, PHI Learning Private Limited, 2012.
2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, "Data Structures and Algorithms David E. Goldberg, "Genetic Algorithm In Search Optimization And Machine Learning" Pearson Education India, 2013.
3. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", 3rd Edition, Pearson Education, 2012.
4. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, 2nd Edition, Universities Press, 2007.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A : 10 Questions of 2 marks each-No choice****20 Marks****PART B : 2 Questions from each unit with internal choice, each carrying 16 marks****80 Marks**

SCSA1501	OPERATING SYSTEMS	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To have an overview of different types of operating systems.
- To learn and implement the concept of process management.
- To observe the concept of storage management.
- To understand the concept of I/O and file systems.
- To learn the basics of Linux Programming.

UNIT 1 INTRODUCTION**9 Hrs.**

Introduction - Operating system structures - System components - OS services - System calls - System structure - Resources Processes - Threads - Objects - Device management - Different approaches - Buffering device drivers.

UNIT 2 PROCESS MANAGEMENT**9 Hrs.**

Processes - Process concepts - Process scheduling - Operations on processes - Cooperating processes - CPU scheduling - Basic concepts - Scheduling criteria - Scheduling algorithms - Preemptive strategies - Non-preemptive strategies.

UNIT 3 SYNCHRONIZATION AND DEADLOCKS**9 Hrs.**

The critical section problem - Semaphores - Classic problems of synchronization - Critical regions - Monitors-Dead locks - Deadlock characterization - Prevention - Avoidance - Detection - Recovery.

UNIT 4 MEMORY MANAGEMENT**9 Hrs.**

Storage Management Strategies - Contiguous Vs. Non-Contiguous Storage Allocation - Fixed and Variable Partition Multiprogramming - Paging - Segmentation - Paging/Segmentation Systems - Page Replacement Strategies - Demand & Anticipatory Paging - File Concepts - Access Methods - Directory Structure - File Sharing - Protection - File - System Structure - Implementation.

UNIT 5 I/O SYSTEM, LINUX & SHELL PROGRAMMING**9 Hrs.**

Mass Storage Structure - Disk Structure- Disk Scheduling - Disk Management - Swap Space Management - RAID Structure - Shell Operation Commands - File Management Operation - Internet Service - Telnet - FTP - Filters and Regular Expressions- Case Study (Linux) - Shell Programming - Variable, Arithmetic Operations, Control Structures, Handling Date, Time and System Information.

Max. 45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Understand the fundamental components of a computer operating system and how computing resources are managed by the operating system.
- CO2 - Apply the concepts of CPU scheduling, synchronization and deadlocks in real computing problems.
- CO3 - Demonstrate the different memory and I/O management techniques used in Operating Systems.
- CO4 - Have practical exposure to the concepts of semaphores and monitors for process synchronization.
- CO5 - Create design and construct the following OS components: Schedulers, Memory management systems in the modern operating system.
- CO6 - Understand file system structure and implement a file system such as FAT.

TEXT/ REFERENCE BOOKS

1. Abraham Silberschatz, Peter Galvin and Gagne, "Operating System Concepts", 10th Edition, Addison Wesley, 2018.
2. Harvey M.Deitel, "Operating System", 3rd Edition, Addison Wesley, 2004.
3. Gary Nutt, "Operating System, A modern perspective", 3rd Edition, Addison Wesley, 2004.
4. Richard Peterson, "Linux : The Complete Reference", 6th Edition, Tata McGraw Hills, 2008.
5. Andrew S. Tanenbaum, "Modern Operating Systems".4th edition 2015.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A :** 10 Questions of 2 marks each-No choice**20 Marks****PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**80 Marks**

SCSA1502	COMPUTER NETWORKS AND DESIGN	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To recognize the principles of the big picture of computer networks.
- To understand the networking environment.
- To know the importance of VPNs.
- To convey the availability of tools and techniques for networking.
- To discuss about evolving technologies in networks.

UNIT 1 FUNDAMENTALS OF NETWORK DESIGN**9 Hrs.**

Design Principles - Determining Requirements - Analyzing the Existing Network - Preparing the Preliminary Design - Completing the Final Design Development - Deploying the Network - Monitoring and Redesigning - Maintaining - Design Documentation - Modular Network Design - Hierarchical Network Design.

UNIT 2 UNDERLYING LAN CONCEPTS**9 Hrs.**

LAN connectivity for small businesses – Integration – Token-Ring – Ethernet – ATM LAN emulation – InterLAN Switching – LAN to Mainframe – Building networks.

UNIT 3 VPNS, INTRANETS AND EXTRANETS**9 Hrs.**

Virtual Network management and planning – VPNs for small businesses – Secure remote access in VPNs – IPSec VPNs – Integrating data centers with Intranets – Implementing and supporting Extranets.

UNIT 4 NETWORKING TOOLS AND TECHNIQUES**9 Hrs.**

Simulation method for designing multimedia networks – Determining remote bridge and router delays – Network baselining as a planning tool.

UNIT 5 EVOLVING TECHNOLOGIES**9 Hrs.**

Trends in data communications – Merits of xDSL technology – Preparing for cable modems - Voice and video on the LAN – Internet voice applications – Building IP PBX telephony network – Fax over IP – Videoconferencing over IP networks.

Max.45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Understand the principles of networks.
- CO2 - Interpret LAN concepts and design.
- CO3 - Gain knowledge in evolving technologies.
- CO4 - Clearly outline the logic behind VPNs.
- CO5 - Know the importance of tools and techniques in building a network.
- CO6 - Understand the underlying working concepts of a real-time network.

TEXT/ REFERENCE BOOKS

1. Gil Held, "Network Design: Principles and Applications (Best Practices)", Auerbach Publications, 1st edition, 2000.
2. Diane Tiare and Catherine Paquet, "Campus Network Design Fundamentals", Pearson Education, 1st edition, 2006.
3. Larry L. Peterson, Bruce S. Davie, "Computer Networks: A Systems Approach", Morgan Kaufmann Publishers Inc., 5th edition, 2012.
4. William Stallings, "Data and Computer Communications", Pearson Education, 8th edition, 2016.
5. James F. Kurose, Keith W. Ross, "Computer Networking - A Top-Down Approach Featuring the Internet", Pearson Education, 6th edition, 2012.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****PART A :** 10 Questions of 2 marks each-No choice**PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**Exam Duration : 3 Hrs.****20 Marks****80 Marks**

SCSA1503	COMPUTER GRAPHICS AND MULTIMEDIA APPLICATIONS	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To introduce two and three dimensional graphical structure.
- To design 2D and 3D methods and Models.
- To explore the visible surface detection with illumination and color models.
- To understand the concepts involved in multimedia and basis tools.
- To gain knowledge on Multimedia compression and animations.

UNIT 1 BASICS OF COMPUTER GRAPHICS**9 Hrs.**

Output primitives- Survey of Computer Graphics-Overview of Graphics System- Line drawing Algorithm (DDA Line Drawing Algorithm, Bresenham's Line Drawing Algorithm)-Circle drawing Algorithm- Curve Drawing Algorithm- Attributes of output Primitives- Antialiasing.

UNIT 2 2D TRANSFORMATION AND VIEWING**9 Hrs.**

2D Transformation and other transformation – 2D and 3D Viewing- Line Clipping(Cohen Sutherland)– Polygon Clipping (Sutherland Hodgeman) – Logical Classification Input Function.

UNIT 3 3D CONCEPTS AND CURVES**9 Hrs.**

3D Object: Representation Method- B-Rep- Sweep Representation- 3D Transformation – curve generation – Splines – Bezier's – Blending of curves other interpolation techniques- Display Curves and Surface – Shape Description Requirements – Parametric function – 3D Concept Introduction – Fractals and Self Similarity – Successive refinement of Curves – Koch Curves and Peano Curves.

UNIT 4 METHODS AND MODELS**9 Hrs.**

Visual Surface detection methods – Illumination models – Halftone Patterns – Dithering Techniques – Polygon Rendering Methods – Ray Tracing Methods – Color methods – Color Applications.

UNIT 5 MULTIMEDIA BASIS AND TOOLS**9 Hrs.**

Multimedia Basics and Tools – Introduction to Multimedia – Compression and Decompression – Data and File Format Standards – Digital voice and audio video image animation- Introduction to photoshop- workshop tools- Navigating window – Importing and Exporting Images – Operations on Images – resize, Crop, rotate.Introduction to Flash – Elements of Flash Documents – Flash Environment- Drawing Tools – Flash Animation Importing and Exporting – Adding Sounds – Publishing Flash Movies.

Max.45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Identify the various line and circle drawing algorithm.
- CO2 - Study the various transformations in 2D and 3D objects.
- CO3 - Understand the concepts of curves and surface.
- CO4 - Apply transformation and clipping algorithm in 2D and 3D objects.
- CO5 - Design Illumination and color models.
- CO6 - Implement 2D and 3D Transformation concepts in Real world Applications.

TEXT / REFERENCE BOOKS

1. Donald Hearn and M. Pauline Baker, Computer Graphics C Version, 2nd Edition, Prentice Hall, 2006.
2. Fabio Ganovelli, Massimiliano Corsini, Sumanta Pattanaik, Marco Di Benedetto "Introduction to Computer Graphics: A Practical Learning Approach" Taylor and Francis Group. 2015.
3. Tay Vaughan, "Multimedia", 5th Edition, Tata McGraw Hill, 2001.
4. Ze-Nian Li, Mark S. Drew, "Fundamentals of Multimedia", Prentice Hall of India, 2004.
5. D.P. Mukherjee, "Fundamentals Of Computer Graphics And Multimedia " Prentice Hall of India Private Limited, 2006.
6. D. McClelland, L.U.Fuller, "Photoshop CS2 Bible", Wiley Publishing, 2005.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****PART A :** 10 Questions of 2 marks each-No choice**PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**Exam Duration : 3 Hrs.****20 Marks****80 Marks**

SCSA1601	MACHINE LEARNING	L	T	P	Credits	Total Marks
		3	*	0	3	100

COURSE OBJECTIVES

- To focus on the construction and study of algorithms that can learn from data.
- To emphasize on the logical, knowledge-based approach.
- To introduce students to the basic concepts and techniques of Machine Learning.
- To develop skills of using recent machine learning software for solving practical problems.
- To gain experience of doing independent study and research.

UNIT 1 INTRODUCTION TO MACHINE LEARNING**9 Hrs.**

Machine learning - examples of machine learning applications - Learning associations - Classification -Regression - Unsupervised learning - Supervised Learning - Learning class from examples - PAC learning -Noise, model selection and generalization - Dimension of supervised machine learning algorithm.

UNIT 2 DECISION THEORY**9 Hrs.**

Bayesian Decision Theory- Introduction- Classification - Discriminant function-Bayesian networks-Association rule - Parametric Methods - Introduction - Estimation -Multivariate methods-Data Parameter estimation-Dimensionality Reduction- PCA-Linear discriminant analysis.

UNIT 3 CLUSTERING AND REGRESSION**9 Hrs.**

Clustering - Mixture densities - k-means clustering - Supervised Learning after clustering - Hierarchical clustering - Nonparametric Methods - Density estimation - Generalization of multivariate data - Smoothing models -Decision Trees - Univariate trees - Multivariate trees - Learning rules from data - Linear Discrimination-Gradient Descent.

UNIT 4 MULTILAYER PERCEPTRONS**9 Hrs.**

Structure of brain - Neural networks as a parallel processing - Perceptron - Multilayer perceptron - Back propagation - Training procedures - Tuning the network size - Learning time.

UNIT 5 LOCAL MODELS**9 Hrs.**

Competitive learning - Adaptive resonance theory - Self organizing map -Radial Basis functions - Bagging- Boosting- Reinforcement Learning.

Max.45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Understand complexity of Machine Learning algorithms and their limitations.
- CO2 - Understand modern notions in data analysis oriented computing.
- CO3 - Be capable of confidently applying common Machine Learning algorithms in practice and implementing their own.
- CO4 - Be capable of performing distributed computations.
- CO5 - Can demonstrate working knowledge of reasoning in the presence of incomplete and/or uncertain information.
- CO6 - Gain ability to apply knowledge representation, reasoning, and machine learning techniques to real-world problems.

TEXT / REFERENCE BOOKS

1. Ethem Alpaydin, "Introduction to Machine Learning", MIT Press, 2004.
2. Tom Mitchell, "Machine Learning", McGraw Hill, 1997.
3. Shai Shalev-Shwartz and Shai Ben-David, "Understanding Machine Learning: From Theory to Algorithms", Cambridge University Press, 2014.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****PART A :** 10 Questions of 2 marks each-No choice**PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**Exam Duration : 3 Hrs.****20 Marks****80 Marks**

SCSA1602	NETWORK SECURITY	L	T	P	Credits	Total Marks
		3	*	0	3	100

COURSE OBJECTIVES

- To understand the fundamentals of Cryptography.
- To acquire knowledge on standard algorithms used to provide confidentiality, integrity and authenticity.
- To explore the various key distribution and management schemes.
- To understand how to deploy encryption techniques to secure data in transit across data networks.
- To learn various mechanisms for network security to protect against the threats in the networks.

UNIT 1 INTRODUCTION**9 Hrs.**

Services, Mechanisms and attacks - The OSI Security Architecture- A Model for Network Security – Classical Encryption Technique – Symmetric Cipher Model – Substitution Technique – Rotor Machines – Steganography.

UNIT 2 BLOCK CIPHERS AND THE DATA ENCRYPTION STANDARD (DES)**9 Hrs.**

Simplified DES- Block Cipher principles – The Data Encryption Standard – The strength of DES – Confidentiality using symmetric encryption – Placement of encryption - Traffic confidentiality – Key distribution - Random number generation.

UNIT 3 PUBLIC KEY ENCRYPTION AND KEY MANAGEMENT**9 Hrs.**

Introduction to number theory – Public key cryptography and RSA – Key Management Diffie-hellman Key exchange.

UNIT 4 AUTHENTICATION AND HASH FUNCTIONS**9 Hrs.**

Authentication requirements – Authentication functions – message authentication codes – Hash functions – Security of hash functions and MAC'S – MD 5 (Message Digest Algorithm) – HMAC.

Digital Signatures and authentication protocols:

Digital Signatures – Authentication protocols – Digital Signature Standard – Kerbews – X.509 Authentication Service.

UNIT 5 NETWORK SECURITY AND SYSTEM SECURITY**9 Hrs.**

Electronic Mail Security – IP Security – Web Security – Intruders – Malicious S/Ws – Firewalls.

Max.45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Implement various symmetric encryption techniques for given applications.
- CO2 - Illustrate various public key encryption techniques.
- CO3 - Understand various key encryption mechanisms and key management strategies that can be applied for real time transactions.
- CO4 - Evaluate authentication and hash algorithms.
- CO5 - Summarize the basic network security mechanisms.
- CO6 - Basic concepts of system level security.

TEXT / REFERENCE BOOKS

1. William Stallings, "Cryptography and Network Security", 6th edition, Pearson Education, 2013.
2. Behrouz A. Forouzan "cryptography and network security", ACM Digital Library, 2007
3. Man Young Rhee, "Internet security: cryptographic principles", "Algorithms and Protocols" Whey publications, 2003
4. Charlie Kaufman, Radia Perlman and Mike Speciner, "Network Security", Prentice Hall of India, 2002.
5. Joey Holland, "Cryptography: Principles and Practice" Larsen and Keller, 2017.
6. Sahadeo Padhye, Rajeev A. Sahu, Vishal Saraswat, "Introduction of Cryptography", CRC press, 2018.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A :** 10 Questions of 2 marks each-No choice**20 Marks****PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**80 Marks**

SCSA1604	COMPILER DESIGN	L	T	P	Credits	Total Marks
		3	*	0	3	100

COURSE OBJECTIVES

- To understand, design and implement a lexical analyzer.
- To understand, design and implement a parser.
- To understand, design code generation schemes.
- To understand optimization of codes and runtime environment.
- To design and develop a compiler.

UNIT 1 INTRODUCTION TO COMPILERS**9 Hrs.**

Structure of compiler – Functions and Roles of lexical phase – Input buffering – Representation of tokens using regular expression – LEX- Properties of regular expression – Finite Automata – Regular Expression to Finite Automata – NFA to Minimized DFA.

UNIT 2 PARSER**9 Hrs.**

Role of Parser-Context-free Grammar – Derivations and Parse Tree - Types of Parser –Bottom Up: Shift Reduce Parsing - Operator Precedence Parsing, SLR parser- Top Down: Recursive Decent Parser - Non-Recursive Decent Parser-Error handling and Recovery in Syntax Analyzer-YACC.

UNIT 3 INTERMEDIATE CODE GENERATION**9 Hrs.**

Types of Intermediate Code – Representation of three address code - Syntax Directed Translation scheme- Intermediate code generation for: Assignment statements - Boolean statements - Switch-case statement –Procedure call - Symbol Table Generation.

UNIT 4 CODE OPTIMIZATION**9 Hrs.**

Principle sources of Optimization - Basic Blocks and Flow Graphs - Loop Optimization and its types – DAG - Peephole optimization - Dominators - Global Data Flow Analysis.

UNIT 5 CODE GENERATION**9 Hrs.**

Issues involved in Code generation – Register allocation – Conversion of three address code to assembly code using code generation algorithm – Examples – Procedure for converting assembly code to machine code – Case study.

Max. 45 Hrs.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 - Describe the various stages involved in the design of a compiler.
- CO2 - Construct grammars and lexical rules for a programming language.
- CO3 - Explain the syntactic and semantic structure in compiler design.
- CO4 - Evaluate various methods of optimizations on intermediate code.
- CO5 - Generate target code for any source code.
- CO6 - Design, develop, and implement a compiler for any programming language.

TEXT / REFERENCE BOOKS

1. Alfred V Aho, Monica S. Lam, Ravi Sethi and Jeffrey D Ullman, "Compilers – Principles, Techniques and Tools", 2nd Edition, Pearson Education, 2007.
2. Steven S. Muchnick, "Advanced Compiler Design and Implementation,"Morgan Kaufmann Publishers – Elsevier Science, India, Indian Reprint 2003.
3. V. Raghavan, Principles of Compiler Design, Tata McGraw Hill Education Publishers, 2010.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks : 100****PART A :** 10 Questions of 2 marks each-No choice**PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**Exam Duration : 3 Hrs.****20 Marks****80 Marks**

SCSA2101	PYTHON AND PROBLEM SOLVING TECHNIQUES LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

COURSE OBJECTIVES

- Identify the problem.
- List the possible ways to obtain the solution.
- Evaluate and Select the Best algorithm to solve the problem.
- Deploy suitable methods to get the desired output.
- Call the methods in order.

SUGGESTED LIST OF EXPERIMENTS

1. Program to exchange the values of two variables.
2. Program to circulate the values of n variables.
3. Program to find distance between two points.
4. Program to find square root.
5. Program to find GCD.
6. Program to find Exponentiation.
7. Program to find sum an array of numbers.
8. Program to find factorial.
9. Program to implement Sine function computation.
10. Program to Generate the Fibonacci sequence.
11. Program for Reversing the digits of an integer.
12. Program to find the smallest divisor of an integer.
13. Program to find the greatest common divisor of two integers.
14. Program to Generate Prime Numbers.
15. Program to Compute the Prime Factors of an integer.
16. Program to Raise a Number to a Large Power.
17. Program for Removal of Duplicates.
18. Program for Partitioning.
19. Program to find the kth smallest Element.
20. Program to generate histogram.

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1 - Analyze and classify the given problem into various modules.
- CO2 - Design the program with basic syntax by reading input from the user and generating the desired output.
- CO3 - Develop the codes containing looping and decision making statements.
- CO4 - Implement user defined functions.
- CO5 - Apply recursion and call the function with appropriate parameters.

SCSA2201	DATA STRUCTURES LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

COURSE OBJECTIVES

- To implement linear and non-linear data structures.
- To understand the different operations of search trees.
- To implement graph traversal algorithms.
- To get familiarized to sorting algorithms.
- To implement linear search and binary Search.

SUGGESTED LIST OF EXPERIMENTS

1. Program to insert and delete an element in an array.
2. Program to implement operations on a Singly linked list.
3. Program to implement operations on a doubly linked list.
4. Program to sort the elements using insertion sort.
5. Program to sort the elements using quick sort.
6. Program to sort the elements using merge sort.
7. Program to implement a Stack using an array and Linked list.
8. Program to implement Queue using an array and Linked list.
9. Program to implement Circular Queue.
10. Program to convert an infix expression to postfix expression.
11. Program to implement display elements of a queue according to their priority.

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1 - Remembering the concept of data structures through ADT including List, Stack and Queues.
- CO2 - Understand basic concepts about stacks, queues, lists, trees and graphs.
- CO3 - Able to apply and implement various tree traversal algorithms and ensure their correctness.
- CO4 - Ability to analyze algorithms and develop algorithms through step by step approach in solving problems with the help of fundamental data structures.
- CO5 - Compare and contrast Array based and Link based applications of typical data structures such as Stacks and Queues.
- CO6 - Design applications and justify use of specific linear data structures for various applications.

SCSA2301	DATABASE MANAGEMENT SYSTEMS LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

COURSE OBJECTIVES

- To instill sound understanding on the fundamentals of DBMS.
- To lay a formal foundation on the relational model.
- To introduce the basic concepts of SQL as a universal database language.
- To understand the necessity for normalization.
- To formulate problem statements for real life problems.

SUGGESTED LIST OF EXPERIMENTS

1. Data Definition Language(DDL).
2. Data Manipulation Language(DML).
3. Data Control Language(DCL).
4. Constraints and Built-in functions.
5. Joins and Group-by functions.
6. PL/SQL programs using functions.
7. PL/SQL programs using procedures.
8. PL/SQL programs using triggers.
9. Developing GUI applications using PHP.
 - Student Information System.
 - Inventory Management.
 - Payroll Processing.

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1 - Understand, appreciate and effectively explain the underlying concepts of database technologies.
- CO2 - Populate and query a database using SQL DML/DDL commands.
- CO3 - Declare and enforce integrity constraints on a database.
- CO4 - Retrieve data from multiple tables.
- CO5 - Programming PL/SQL including stored procedures, stored functions, triggers.
- CO6 - Design and build a GUI application using PHP.

SCSA2302	CODE OPTIMIZATION AND DEBUGGING - I	L	T	P	Credits	Total Marks
		0	0	2	1	50

COURSE OBJECTIVES

- To improve the intermediate code by making it consume fewer resources.
- To get the faster running machine code.
- To improve the consistency of the code.
- To enhance the readability of the code and easier code maintenance.
- To improve the work flow of the code.

SUGGESTED LIST OF EXPERIMENTS

1. Implement Built in Function and Libraries using python.
2. Implement Optimizing loop using python.
3. Profiling CPU usage using python.
4. Profiling memory usage using python.
5. Implement query optimization in DBMS.
6. Implement heuristics in query optimization in DBMS.
7. Implement factorial of given number using memorization in data structure.
8. Implement Divide and conquer method in data structure.
9. Implement hybrid stable sorting algorithm in data structure.
10. Implement linked list with improving the time complexity in data structure.

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1: Understand importance of code optimization.
- CO2: Apply time complexity and space complexity of the algorithm for improving optimization.
- CO3: Analyse and appreciate variety of performance measures for various optimization problems.
- CO4: Learn efficient computational procedures to solve optimization problems.
- CO5: Be able to use programming languages to implement optimization algorithms.
- CO6: Be able reduce the execution time of code by applying proper coding technique.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN

Max. Marks: 100

CAE: Evaluation of Regular Lab class

Model practical exam

ESE: End Semester Practical Exam

Exam Duration:3 Hrs.

15 Marks

10 Marks

25 Marks

25 Marks

SCSA2303	IoT LAB	L	T	P	Credits	Total Marks
		0	0	2	1	100

COURSE OBJECTIVES

- To identify the different technologies and different platforms in IoT
- To understand how to use sensors and actuators for design of IoT.
- To learn different protocols used in IOT.
- To learn the concepts of smart city development in IOT.
- To learn how to analysis the data in IOT.

SUGGESTED LIST OF EXPERIMENTS

1. Monitor the soil moisture level using Raspberry pi.
2. Monitor the surrounding air's quality using Arduino & sensors connected to a Raspberry Pi. 3. Implement Raspberry Pi as a local server by installing Apache, PHP, and MySQL.
3. Interface PIR sensor with ESP8266 WiFi module on NodeMCU platform.
4. Posting temperature and humidity of the environment to Thing speak cloud service using NodeMCU.
5. Implement Raspberry Pi Temperature System with Ubidots cloud service.
6. Implement smart cities Sensor Board v3.0 using Wasp mote platform for measuring temperature, humidity and pressure.
7. Measure light intensity and pollution level using Wasp mote platform.
8. Receiving and storing data in Meslium gateway for Wasp mote platform.
9. Implement barometric pressure monitoring using BMP280 sensor for Zolertia module.

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1:** Remember the concept of IoT components for implementing real time applications
- CO2:** Understand basic concepts about the hardware kits Auduino & Raspberry Pi and supported software.
- CO3:** Apply and implement various IoT applications like monitoring air quality, soil moisture etc.
- CO4:** Analyze IoT applications connected with Cloud servers
- CO5:** Design an IoT System for integrating smart city services.
- CO6:** Create an IoT system for real time monitoring applications.

SCSA2401	OBJECT ORIENTED ANALYSIS AND SYSTEM ENGINEERING LAB	L	T	P	Credits	Total Marks
		0	0	2	1	100

COURSE OBJECTIVES

- To understand the object oriented concepts.
- To understand the concept of UML Diagrams.
- To train in Rational ROSE and Agro UML Tool.
- To capture the requirements specification for an intended software system.
- To draw the UML diagrams for the given specification.
- To map the design properly to code.

SUGGESTED LIST OF EXPERIMENTS

1. Study experiment – Object Oriented Concepts.
2. Study of UML diagrams – aim and scope of diagrams.
3. Study of CASE tools – Agrium and Rational ROSE.
4. Draw standard UML diagrams using an UML modelling tool for a given case study and map design to code
 - Identify a software system that needs to be developed.
 - Document the Software Requirements Specification (SRS) for the identified system.
 - Identify the actors and use cases and develop the Use Case model.
 - Identify the classes and develop a Class Diagram by specifying the attributes and methods. The Relationships between the classes are identified to indicate how the classes are related to each other.
 - Using the identified scenarios, find the interaction between objects and represent them using UML Sequence and Collaboration Diagrams.
 - Draw relevant State Chart and Activity Diagrams for the same system.
 - Generate the Code Sequence for
 - a. Student Information System
 - b. Online Ticket Reservation system
 - c. Employee Payroll system
 - d. Online Banking Application
 - e. ATM processing
 - f. Stock Maintenance
 - g. Library Management
 - h. Exam registration
 - i. Book bank

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1 - Identify the Requirement specification of a given problem.
- CO2 - Apply the concept of UML diagrams to perform analysis and design using the Tools.
- CO3 - Perform OO analysis and design for a given problem specification.
- CO4 - Generate the code.
- CO5 - Test the compliance of the software with the SRS.

SCSA2402	CODE OPTIMIZATION AND DEBUGGING - II	L	T	P	Credits	Total Marks
		0	0	2	1	50

COURSE OBJECTIVES

- To analyze the programming logic and apply the appropriate design approach.
- To optimize the speed of the program by using proper utilization of available memory.
- To make use of performance tools for tracking the application.
- To have a capacity to analyze and design software systems, components to meet desired needs.
- To have a working ability and grasping attitude to design and conduct object-oriented analysis.

LIST OF EXPERIMENTS

1. Implement minimum heap allocation in java.
2. Implement string builder in java.
3. Implement concurrency control using java.
4. Checking the current log level using java.
5. Implement Garbage collection using java.
6. Implement grid line in Object oriented analysis and design.
7. Implement grid with two sets of diagonal lines in Object oriented analysis and design.
8. Implement concurrency in Object oriented analysis and design.
9. Implement design optimization for removing non usable associations in Object oriented analysis and design.
10. Implement sparing and storing derived attributes associations in Object oriented analysis and design.

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1: Understand the impact of performance hits during application runtime.
- CO2: Implement optimal functions for improving the performance.
- CO3: Analyze the proper utilization of memory for code.
- CO4: Design experiments using UML, as well as to analyze and evaluate their models.
- CO5: Discussing and understanding analysis and design heuristics that are involved in the course.
- CO6: Understand the map of diagrammatic notations into another.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN

Max. Marks: 100

Exam Duration: 3 Hrs.

CAE: Evaluation of Regular Lab class

15 Marks

Model practical exam

10 Marks

25 Marks

ESE: End Semester Practical Exam

25 Marks

SCSA2404	ROBOTICS LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

COURSE OBJECTIVES

- To introduce different types of robotics and demonstrate them to identify different parts and components.
- To learn the fundamentals of Robot program to perform various operations
- To model and simulate a Robot with pick and place operations.
- To Map the simulated robot into real world applications
- To Design a simple robot to recognize the individual components to assemble industrial applications.

LIST OF EXPERIMENTS

1. Determination of maximum and minimum position of links.
2. Verification of transformation (Position and orientation) with respect to gripper and world coordinate system
3. Estimation of accuracy, repeatability and resolution.
4. Robot programming and simulation for pick and place
5. Robot programming and simulation for Color identification
6. Robot programming and simulation for Shape identification
7. Robot programming and simulation for machining (cutting, welding)
8. Robot programming and simulation for writing practice
9. Robot programming and simulation for any industrial process (Packaging, Assembly)
10. Robot programming and simulation for multi process.

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1:** Remember the concept of Robotics programming
- CO2:** Understand basic concepts about Robot operations
- CO3:** Apply and implement various Robot operations through simulation
- CO4:** Analyze Robot programs for industrial applications
- CO5:** Design a Robot to perform machining operations.
- CO6:** Create an AI Robot to perform the task navigation, sense and calculate their reaction.

SCSA2501	COMPUTER NETWORKS LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

COURSE OBJECTIVES

- To understand how to implement socket programming.
- To be familiar with simulation tools.
- To understand how to create applications using TCP and UDP.
- To gain Knowledge on various networking protocols.

SUGGESTED LIST OF EXPERIMENTS

1. To find IP address of a machine.
2. Time and Date server.
3. Echo UDP server.
4. TCP sockets.
 - a. Echo client & server
 - b. Chat
 - c. File transfer
5. Daemon program.
6. Implementing stop and wait protocol & sliding window protocol.
7. Code simulating ARP/RARP protocols.
8. Code simulating PING and TRACEROUTE commands.
9. RPC (Remote Procedure Call).
10. Study of Network Simulator (NS) and simulation of Congestion Control algorithms using NS.
11. Case study
 - a. Flooding
 - b. Link state routing
 - c. Distance vector routing

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1 - Analyze the performance of the protocols in different layers.
- CO2 - Implement various protocols.
- CO3 - Design with simulation tools.
- CO4 - Analyze various routing algorithms.
- CO5 - Construct Wi-Fi model.
- CO6 - Understand socket programming.

SCSA2502	OPERATING SYSTEMS LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

COURSE OBJECTIVES

- To write programs in Linux environment using system calls.
- To implement the scheduling algorithms.
- To implement page replacement algorithms.
- To implement file allocation methods.
- To develop solutions for synchronization problems using semaphores.

SUGGESTED LIST OF EXPERIMENTS

1. Study of basic LINUX and vi Editor command.
2. String and Numerical Handling Functions.
3. Loop and Selection Constructs.
4. File Handling Functions.
5. Manipulate Date/Time/Calendar.
6. Retrieve System information.
7. Implementation of process scheduling mechanism – FCFS, SJF, Priority Queue.
8. Producer Consumer Problem using Semaphores.
9. Reader – Writer Problem.
10. Diner's Philosopher Problem.
11. First Fit, Worst Fit, Best Fit allocation strategy.
12. Bankers Algorithm.
13. Simulate Paging Technique of Memory Management.
14. Simulate Page Replacement Algorithms.
15. Implement memory management Scheme.

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1 - Learn the basic services and functionalities of the operating system using System Calls in Linux.
- CO2 - Model CPU Scheduling such as FCFS, SJF, Priority and Round Robin.
- CO3 - Describe and solve Synchronization Problem.
- CO4 - Simulate Memory Management Technique.
- CO5 - Explain how to perform Paging Techniques.
- CO6 - Demonstrate Page Replacement Algorithms.

SCSA2602	COMPILER DESIGN LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

COURSE OBJECTIVES

- To gain in depth knowledge, understanding of compiler design.
- To enrich the knowledge in various phases of compiler and its use.
- To extend the knowledge of parser by parsing LL parser and LR parser.
- To provide practical programming skills necessary for constructing a compiler.
- To implement NFA and DFA from a given regular expression.

SUGGESTED LIST OF EXPERIMENTS

1. Use LEX tool to implement Lexical Analyzer.
2. Use LEX and YACC to implement Parser using Ambiguous Grammar.
3. Use LEX and YACC to implement Parser using Unambiguous Grammar.
4. Use LEX and YACC tool to implement Desktop Calculator.
5. Implement Recursive Descent Parser algorithm.
6. Implement Shift Reduce Parser algorithm.
7. Implement Operator Precedence Parser algorithm.
8. Implement the backend of the compiler to produce three address codes.
9. Implement Symbol Table Management.
10. Construction of NFA and DFA from a regular expression.
11. Implementation of simple code optimization techniques.
12. Construct a Simple compiler.

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1 - Apply the LEX tool and YACC tool to develop a scanner and parser.
- CO2 - Demonstrate the process of lexical analysis, parsing and other compiler design aspects.
- CO3 - Achieve practical programming skills necessary for constructing a compiler.
- CO4 - Interpret the symbol table and intermediate code generation.
- CO5 - Construct Automata for any regular language.
- CO6 - Test and simulate the compiler system.

SCSA3091	SOCIETY 5.0	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVE

- To understand the society 5.0, Cyberspace And Physical Space to solve
- To Provide knowledge and overview about Big data, IoT and Artificial Intelligence for Society 5.0.
- Discuss To understand Augmented Reality and Virtual Reality, Next Generation Sensors
- To discuss about Challenges and Technologies towards Society 5.0, Security of Cyber Physical Systems
- Discuss to apply society 5.0 Innovation with Future Trends with Applications

UNIT I**(9 hrs)****INTRODUCTION TO SOCIETY 5.0**

Introduction –Schema of society 5.0-Characteristics of Society 5.0. Introduction to communication technologies: Artificial Intelligence – robotics - 3D Printing. People: Centric Society -Knowledge Sharing- Physical space-Cyberspace – Humanity VS Society 5.0 –Elements of Society 5.0-Data Driven to Society- Modeling real world Issues.

UNIT II**(9 hrs)****EMERGING TECHNOLOGIES WITH SOCIETY 5.0**

Introduction to Big Data – Issues and Challenges in the traditional systems –Intelligent Data Analysis –Big Data Storage Statistical Concepts: Sampling Distributions - Re-Sampling - Prediction Error – Random Sampling– Artificial Intelligence – - Foundations of AI - Intelligent agent - Types of agents - Structure - Problem solving agents -Internet of Things- Introduction to IoT- Basic Architecture of an IoT, From M2M to IoT, M2M towards IoT-Robotics- Robotics system components - Robot classification Coordinate frames - degree of freedom - dynamic stabilization of robots.

UNIT III**(9 hrs)****INTRODUCTION TO INDUSTRY 4.0**

Introduction-Globalization and Emerging Issues, LEAN Production Systems, Smart and Connected Business Perspective, Cyber Physical Systems and Next Generation Sensors, Augmented Reality and Virtual Reality, Artificial Intelligence, Big Data and Advanced Analysis- An emerging industrial structure for IoT -Cyber security in Industry 4.0-Basics of Industrial IoT. Common Issues in Industry 4.0 and Society 5.0.

UNIT IV**(9 hrs)****CHALLENGES AND TECHNOLOGIES TOWARDS SOCIETY 5.0**

Overcome with Economic Development and Solution to Social Problems in Society 5.0- Security of Cyber Physical Systems -Embedded and CPS security - attacks and countermeasures, authentication, identification, confidentiality, data integrity, authorization, access control, malware attacks and counter-measures, security protocols- Social Issues in Society 5.0 - human-centered society (Society 5.0)- Sustainable Development Goals-Economic Advancement- Resolution to Social Problems..

UNIT V**(9 hrs)****INNOVATION WITH FUTURE TRENDS WITH APPLICATIONS**

Mobility – Health Care –Agriculture- Food Products – Disaster Prevention.

Max 45hrs**COURSE OUTCOMES**

- CO1 - Able to understand the Social Problems By A System That Highly Integrates Cyberspace and Physical Space to solve
- CO2 - Ability to get skills For Economic Development And A Human-Centered Society That Balances Economic Advancement
- CO3 - Achieve A High Degree Of Convergence Between Cyberspace (Virtual Space) And Physical Space (Real Space)
- CO4 - Able to Use Of Emerging Technologies With Society 5.0 To Achieve The More Production / Avoid Loss Of Productions
- CO5 - Able Internet and IoT , Big data for production lines to be adaptive, intelligent, and flexible enough to meet the updated requests.
- CO6 - Design and apply in Health Care , Agriculture, Food Products, Disaster Prevention

TEXT/ REFERENCE BOOKS:

1. Society 5.0 A People –Centric Super –Smart Society, Hitachi –Utokyo Laboratory, Springer,2020.
2. Society 5.0 Industry of the Future Technologies Methods and Tools By Bruno Salgues,Wiley, 2018
3. Stuart J.Russel, Peter Norvig, “Artificial Intelligence A Modern Approach ”, 3rd Edition, Pearson Education, 2009.

4. The Internet of Things: Applications and Protocols, Wiley publications. Author(s): Oliver Hersent, David Boswarthick, Omar Elloumi.,2012.
5. McKerrow P.J. "Introduction to Robotics", Addison Wesley, USA, 1991.

SCSA5101	ADVANCED DATA STRUCTURES AND ALGORITHMS	L	T	P	Credits	Total Marks
		3	*	0	3	100

COURSE OBJECTIVES

- To understand and analyze fundamental data structures, such as binary search trees, disjoint sets, and self-adjusting lists.
- Understand the implementation and complexity analysis of fundamental algorithms such as RSA, primarily testing, max flow, discrete Fourier transform.
- Will be exposed to algorithmic issues in a variety of areas, including linear programming and game-theory.

UNIT 1 INTRODUCTION TO ALGORITHMS**9 Hrs.**

Role of Algorithms in computing- Analyzing algorithm- Designing algorithm- Asymptotic Notations- Summations -Formulas and properties- Recurrences. SORTING TECHNIQUES: Heap sort – Quick sort-Radix sort- Bucket sort- Analysis of sorting algorithms.

UNIT 2 TREE STRUCTURES**9 Hrs.**

Binary Search Trees- AVL Trees- Red-Black Trees- B-Trees-Splay Trees - HEAP STRUCTURES: Min/Max heaps- Leftist Heaps- Binomial Heaps- Fibonacci Heaps.

UNIT 3 MULTIMEDIA STRUCTURES**9 Hrs.**

Segment Trees- k-d-Trees-Point Quad Trees-MX-Quad Trees-R-Trees-TV-Trees.

UNIT 4 PROBLEM SOLVING TECHNIQUES**9 Hrs.**

Branch and Bound-NP hard and NP complete problems-Huffman Coding-Activity Networks-Flow Shop Scheduling- Randomized Algorithms. ALGORITHMS: Greedy Algorithms-Backtracking-Dynamic programming- Divide and Conquer.

UNIT 5 GRAPH ALGORITHMS**9 Hrs.**

Graph Representation-BFS-DFS-Topological Sort-Connected Components-Minimum Spanning Trees-Kruskal's Algorithm - Prim's Algorithm – Dijkstra's Algorithm – Floyd's Algorithm – Bellman Ford Algorithm.

Max. 45 Hrs.**TEXT / REFERENCE BOOKS**

1. Thomas H Cormen, Charles E. Leiserson, Ronald L. Rivest, "Introduction to Algorithms", PHI, 2002.
2. Sara Baase, Allen R. Gold, "Computer Algorithms and Introduction to Design and Analysis", Pearson, 2000.
3. Sahni, "Data Structures algorithm and application in C++", PHI, 2000.
4. Mark Allen Weiss, "Data Structures and Algorithms in C++", Addison Wesley, 3rd Edition.
5. Aho A.V., Hopcroft J.E. and Ullman J.D., "Design and analysis of Computer algorithms".

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks: 100****Exam Duration: 3 Hrs.****PART A:** 5 Questions of 6 Marks each – No choice**30 Marks****PART B:** 2 Questions from each unit of internal choice, each carrying 14 Marks**70 Marks**

SCSA5102	DATABASE TECHNOLOGIES	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To give students in depth information about system implementation techniques, data storage, representing data elements, database system architecture.
- Demonstrate principles of design, development and administration relevant to Oracle database technology.
- Formulate a working definition of database development and administration.
- To understand the emerging databases like Mobile, Multimedia, Cloud and Big Data and Hadoop infrastructures.

UNIT 1 INTRODUCTION**9 Hrs.**

Data base system Architecture – Query Optimization Techniques - Transaction Management: Transaction Processing Concepts - Concurrency Control - Recovery Techniques -Database Security.

UNIT 2 PARALLEL DBMS**9 Hrs.**

Architecture-Query evaluation-Query optimization-parallelizing Individual operations. Distributed DBMS: Architecture-storing data-Cataloguing-Query processing-Transactions Concurrency and Recovery.

UNIT 3 MOBILE DATABASES**9 Hrs.**

Mobile Databases: Location and Handoff Management - Effect of Mobility on Data Management - Location Dependent Data Distribution - Mobile Transaction Models -Concurrency Control - Transaction Commit Protocols

UNIT 4 OBJECT AND MULTIMEDIA DATABASE SYSTEMS**9 Hrs.**

Object Databases–Advantages and disadvantages compared to Relational Databases- Abstract data types, Objects identity and reference types- Inheritance Database design for ORDBMS ODMG data model and ODL OQL.MULTIMEDIA DATABASES: Nature of Multimedia data and applications Data management issues– Components of Multimedia database management system.

UNIT 5 BIGDATA AND HADOOP**9 Hrs.**

Big Data–Introduction-Technologies–Reference Architecture- Hadoop– Introduction to Hadoop Distributed File System–Design of HDFS–HDFS Concepts–Interfaces for Hadoop File System–Map Reduce–weather Dataset–Analyzing the data with Hadoop–Anatomy of Map Reduce Job Run.

Max. 45 Hrs.**TEXT / REFERENCE BOOKS**

1. Raghu Ramakrishnan and Johannes Gehrke, "Database Management Systems", 3rd Edition McGraw Hill Publications.
2. Korth.H.F. and A.Silberschatz, "Data base system concepts", 3rd Edition, McGraw Hill Publications.
3. Ramez Elmasri and B.Navathe:"Fundamentals of Database Systems", 5th Ed., Addison Wesley, 2008.
4. Stenfo Ceri and Gieusppepelagatti, "Distributed database-principles and systems", 3rd Edition.
5. O'Reilly, "Hadoop–The Definitive Guide", O'Reilly Media.
6. <http://www.oracle.com/technetwork/topics/entarch/articles/info-mgmt-big-data-ref-arch-1902853.pdf>
7. <http://www.comptechdoc.org/independent/database/basicdb/dataobject.html>.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks: 100****Exam Duration: 3 Hrs.****PART A:** 5 Questions of 6 Marks each – No choice**30 Marks****PART B:** 2 Questions from each unit of internal choice, each carrying 14 Marks**70 Marks**

SCSA5103	SOFTWARE QUALITY ASSURANCE AND TESTING	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To introduce the basics and necessity of software testing.
- To introduce various testing techniques along with software production.
- To introduce the concepts of Software quality and its assurance.

UNIT 1 INTRODUCTION**9 Hrs.**

Software testing fundamentals-objectives-principles-Testability-Levels of testing-TMM- Characteristics of a tester-A testing group-Comparison of KPA in ISO,CMM and TMM-Software quality attributes.

UNIT 2 SOFTWARE TESTING METHODS AND STRATEGIES**9 Hrs.**

Test case design-White box testing- Basic path testing-Flow Graph Notation-Cyclomatic Complexity-Deriving Test Cases-Graphic matrices-Control Structure Testing-Deriving Test cases-Dataflow Testing-Loop Testing-Black box Testing-Graph Based Testing Methods-Equivalence partitioning-Boundary value analysis-Comparison testing. Strategic Approach to Software Testing-Verification and Validation-Strategic issues-Unit testing-Consideration Procedures. Integration Testing-Regression Testing- Integration Test Documentation-Validation testing-Configuration Review-Alpha and Beta Testing System Testing-Art of Debugging. Case studies: Writing test cases for white box and black box testing.

UNIT 3 OBJECT ORIENTED TESTING, METRICS AND MEASUREMENTS**9 Hrs.**

Testing OOD Models-OO Testing Strategies-Test Case Design for OO Software-Testing Methods applicable at the class level-Inter class test design-The intent of OO metrics-Distinguishing characteristics of OO metrics. Common software measurements: Code coverage-Cohesion-Comment density, Coupling-Function point analysis- Number of classes and interfaces-Number of lines of customer requirements-Robert Cecil Martin's software package metrics-Bugs per line of code-Source lines of code.

UNIT 4 QUALITY CONSIDERATIONS**9 Hrs.**

Planning for quality – Quality plan – Cost/benefit tradeoff – Feature/bug tradeoff – Modeling quality improvement-Notation of defects-Defects-Defect removal activities-Conditions for each activity-Defect removal goals-Quality improvement teams-quality recognition-Manage Priorities Effectively-Statistical Quality Assurance-Software Reliability.

UNIT 5 OVERVIEW OF STANDARDS**9 Hrs.**

Levels of standards-Quality assurance standards-Project management standards-Dependability standards-Product standards-Process standards-ISO9000-Reviews,Audits and Certification.

Max. 45 Hrs.**TEXT / REFERENCE BOOKS**

1. Ilene Barnstein,"Practical Software Testing-A Process Oriented Approach", Springer International Edition.
2. Humphrey, W.S. (2002), "A Discipline for Software Engineering. Reading, MA", Addison-Wesley.
3. Roger S.Pressman, "Software Engineering, A Practitioner's Approach", Tata McGraw Hill, 5th Edition.
4. Marc Roper, "Software Testing", McGraw Hill Professional, 1994.
5. Ian Sommerville, "Software Engineering", Addison-Wesley, 5th Edition, 1996.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks: 100****Exam Duration: 3 Hrs.****PART A:** 5 Questions of 6 Marks each – No choice**30 Marks****PART B:** 2 Questions from each unit of internal choice, each carrying 14 Marks**70 Marks**

SCSA5104	NETWORK TECHNOLOGIES AND PROTOCOLS	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To understand how routing is done in telephone networks.
- To learn about the different internet routing protocols.
- To appreciate the different aspects routing in optical and mobile networks.
- To understand the issues in adhoc networks, the protocols used for the working of adhoc networks.

UNIT 1 INTRODUCTION**9 Hrs.**

ISO OSI Layer Architecture-TCP/IP Layer Architecture – Functions of Network layer-General Classification of routing-Routing in telephone networks-Dynamic Non Hierarchical Routing(DNHR)-Trunk status map routing (TSMR)-Real-time network routing(RTNR)-Distance vector routing-Link state routing-Hierarchical routing.

UNIT 2 WIRED LINE AND WIRELESS NETWORKS**9 Hrs.**

LAN IEEE802 Projects-Ethernet-Token Ring-FDDI-MAN-DQDB-SMDS-Wireless LAN-IEEE 802.11-WiFi-SONET-DWDM-DSL-Switching-Circuit Switching-Packet switching Intelligent Networks-LAN bridges.

UNIT 3 INTERNET ROUTING PROTOCOLS**9 Hrs.**

Interior routing protocols: Routing Information Protocol (RIP) - Open Shortest Path First (OSPF) – Exterior Routing protocols: Exterior Gateway Protocol(EGP) and Border Gateway Protocol(BGP).Multicast routing :Pros and cons of Multicast and Multiple Unicast Routing - Distance Vector Multicast Routing Protocol (DVMRP) - Multicast Open Shortest Path First(MOSPF) MBONE.

UNIT 4 ADVANCED NETWORKS CONCEPTS**9 Hrs.**

VPN-Remote-Access VPN-site-to-site VPN-Tunneling to PPP-Security in VPN-MPLS-Operation– Routing - Tunneling and Use of FEC - Traffic Engineering and MPLS based VPN - Overlay Networks - P2P Connections - IPv4 vsIPv6.

UNIT 5 TELE COMMUNICATION AND CELLULAR NETWORKS**9 Hrs.**

Telecommunication systems-GSM-Services-Architecture-Protocols-Handover-DECT-Architecture-Protocols-Handovers-TETRA- Bluetooth Technology-GPRS-WAP-WAP Architecture-WAP Protocols- WLL.

Max. 45 Hrs.**TEXT / REFERENCE BOOKS**

1. William Stallings, "High-Speed Networks and Internets, Performance and Quality of Service", 2nd Edition, Pearson Education Asia, Reprint India, 2002.
2. Martha Steens trup, "Routing in Communication Networks", Prentice Hall International, New York, 1995.
3. Leon Garcia.Widjaja,"Communication Networks", Tata McGraw-Hill, 6th Edition, 2002.
4. Aunurag Kumar, D.Manjunath, Joy Kuri, "Communication Networking", Morgan Kaufmann Publishers, 2011.
5. Jochen Schiller, "Mobile Communications", 2nd Edition, Pearson Education, 2008.
6. Asoke K Talukder, Hasan Ahmed, Roopa R Yavagal, "Mobile Computing: Technology, Applications and Service Creation", 2nd Edition, Tata McGraw Hill, 2010.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks: 100****Exam Duration: 3 Hrs.****PART A:** 5 Questions of 6 Marks each – No choice**30 Marks****PART B:** 2 Questions from each unit of internal choice, each carrying 14 Marks**70 Marks**

SCSA5105	ADVANCED OPTIMIZATION TECHNIQUES	L	T	P	Credits	Total Marks
		3	*	0	3	100

COURSE OBJECTIVES

- To learn linear and non linear programming problem.
- To understand the concept of queuing model, simulation and decision theory.

UNIT 1 INTRODUCTION TO OPTIMIZATION 9 Hrs.

Formulation of optimization problem- Simplex Method - Big M Method - Two phase method - Dual Simplex method - Integer Programming- Gomory All Integer cutting plane Method-Gomory Mixed Integer Method-Branch and Bound Method

UNIT 2 NON LINEAR PROGRAMMING 9 Hrs.

Introduction – Unconstrained and Constrained Optimization- Kuhn Tucker conditions- Relative Maximum and Minimum values- Method of Lagrangian Multipliers- Hessian Matrix- Quadratic programming- Wolfe's Modified Simplex Method – Problems

UNIT 3 DYNAMIC PROGRAMMING 9 Hrs.

Recursive relationship - Solution to recursive equation - Dynamic Programming Algorithm - Principle of Optimality - Maximum and minimum values - Solution of LPP by Dynamic Programming - Multi stage problem

UNIT 4 QUEUEING NETWORKS 9 Hrs.

Introduction to Concept of Queuing Models - Single Server - Multiple server Models - Problems - Pollaczek Khinchine theorem. Theoretical concepts of Open queuing networks (Theory) - Closed Queuing Networks (Theory) - Queues in series (Theory).

UNIT 5 DECISION ANALYSIS AND SIMULATION 9 Hrs.

Introduction to Decision Making process – Elements – Decision making under uncertainty – Maximin and Maximax criteria- Hurwitz criterion – Laplace criterion – Minimax Regret criterion – Decision tree analysis- Problems - Simulation - Nature and need for simulation - Monte Carlo method – Applications to Queuing systems.

Max. 45 Hrs.**TEXT / REFERENCE BOOKS**

1. J.K. Sharma, "Operations Research Theory and Applications", 4th Edition, Macmillan India Ltd., 2010.
2. F.S.Hillier & G.J. Lieberman, "Introduction to Mathematical Programming", McGraw Hill International Edition.
3. H.A.Taha "Operations Research: An Introduction", 6th Edition, Macmillan.
4. Prem Kumar Gupta and D.S. Hira, "Operations research", S Chand, 2000.
5. Sharma .S.D, "Operations Research", Kedarnath Ramnath & Co, 2002.
6. Gupta. P.K and Man Mohan "Problems in Operations Research", Sultan Chand and Sons, 7th Edition, 1995.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks: 100****Exam Duration: 3 Hrs.****PART A:** 5 Questions of 6 Marks each – No choice**30 Marks****PART B:** 2 Questions from each unit of internal choice, each carrying 14 Marks**70 Marks**

SCSA5301	INTERNET OF THINGS	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To understand the concept of Internet of Things.
- To identify the various elements of an IoT System.
- To understand the various means of communication from Node / Gateway to Cloud Platforms.
- To understand Cloud Computing & its relevance in IoT.
- To identify types of data analytics and data visualization tools.
- To make students aware of security concerns and challenges while implementing IoT solutions.

UNIT 1 INTRODUCTION TO IOT**9 Hrs.**

Introduction to IoT - Current technological trends and future prospects - Evolution of IoT- Business Scope - Relation with embedded system - Basic Architecture of an IoT - From M2M to IoT - M2M towards IoT - IoT Value Chains - An emerging industrial structure for IoT.

UNIT 2 ELEMENTS OF IOT**9 Hrs.**

Application Sensors and Actuators - Edge Networking (WSN) – Gateways - IoT Communication Model – WPAN and LPWA - Overview of IoT supported Hardware platforms such as: Raspberry pi - ARM Cortex Processors - Arduino and Intel Galileo boards - Wearable Development Boards.

UNIT 3 COMMUNICATION AND CONNECTIVE TECHNOLOGIES**9 Hrs.**

IoT Communication Model - Cloud computing in IoT - IoT in cloud architecture - Logging on to cloud - Selecting and Creating cloud service - cloud based IoT platforms - IBM Watson - Google cloud.

UNIT 4 DATA ANALYTICS AND IOT PLATFORM**9 Hrs.**

Big Data Analytics - Apache Hadoop - Using Hadoop Map Reduce for Batch Data Analysis - Apache Storm - Data Visualization - Visualization tools for IoT.

UNIT 5 HANDS-ON PROJECTS**9 Hrs.**

Industry 4.0 concepts - Sensors and sensor Node and interfacing using any Embedded target boards (Raspberry Pi / Intel Galileo/ARM Cortex/ Arduino) - DIY Kits – Soil moisture monitoring - Weather monitoring - Air quality Monitoring - Movement Detection.

Max. 45 Hrs.**TEXT / REFERENCE BOOKS**

1. The Internet of Things: Applications and Protocols, Wiley publications. Author(s): Oliver Hersent, David Boswarthick, Omar Elloumi.
2. Architecting the Internet of Things, Springer publications. Author(s): Dieter Uckelmann, Mark Harrison, Florian Michahelles.
3. Internet of Things with Arduino Cookbook, Packt Publications. Author(s): Marco Schwatz.
4. Internet of Things and Data Analytics, Wiley Publications.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks: 100****Exam Duration: 3 Hrs.****PART A:** 5 Questions of 6 Marks each – No choice**30 Marks****PART B:** 2 Questions from each unit of internal choice, each carrying 14 Marks**70 Marks**

SCSA5302	GREEN COMPUTING	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To study the existing green computing strategies, fundamental challenges in achieving green operations of computing units and directions to solve some of them.
- To acquire expertise for improving the energy efficiency of personal computers by reducing the power consumption requirements.
- To study the regulatory and governance issues surrounding IT industry.

UNIT 1 GREEN COMPUTING FUNDAMENTALS**9 Hrs.**

Information Technology and Environment - Business and Environment - Green Enterprise Characteristics- Green Vision- Green Strategic Points-Green Value-Green IT Opportunity-Environmental Intelligence-Business Intelligence-Envisioning the Green Future.

UNIT 2 GREEN IT STRATEGIES AND ASSETS**9 Hrs.**

Introducing Green IT Strategies- Green IT Drivers- Green IT Business Dimensions- Green IT Metrics and Measurements- Green IT Readiness and CMM- Green Assets- Buildings- Green IT Hardware- Green Data Centers- Data servers Optimization and Virtualization- cloud Computing and Data Centers- Networking and Communication Infrastructure- Managing Devices for Central Green Services.

UNIT 3 SOCIOCULTURAL ASPECTS OF GREEN IT**9 Hrs.**

Green IT's Social Impact – Green Socials take holders – Role based view of Green IT- Green User practices- Attitude and Subjectivity in Green IT- Green IT Ethics and Code Conduct – Privacy and security of green Information -Green IT project - Green HR and Changing Organization Structures - Green Virtual Communities.

UNIT 4 EMERGENT CARBON ISSUES-TECHNOLOGIES AND FUTURE**9 Hrs.**

Future Carbon Landscape - Green ICT and Technology Trends - Cloud Computing – SaaS – Nanotechnologies- Quantum Computing – Eco design – Biomimicry - New Renewable Energies – Green ICT- Business and Economic Trends.

UNIT 5 CASE STUDIES**9 Hrs.**

Applying Green IT Strategies and Application to a Hospital -Packing Industry and Telecom Sector.

Max. 45 Hrs.**TEXT / REFERENCE BOOKS**

1. Bhuvan Unhelkar, "Green IT Strategies and Applications: Using Environmental Intelligence", CRC Press, 2011 ISBN 978-1- 4398-3760.
2. Toby Velte, Anthony Velte, Robert Elsenpeter, "Green IT-Reduce Your Information System's Environmental Impact While Adding to the Bottom Line "First Edition, Anthony T. Veleto, Robert Elsenpeter, Publisher: McGraw-Hill, 2008, ISBN: 978-0- 07-159923-8, Pages: 307.
3. John Lamb, "The Greening of IT-How Companies Can Make a Difference for the Environment", Publisher: IBM Press, 2009, ISBN: 978-0-13-715083-0.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN**Max. Marks: 100****Exam Duration: 3 Hrs.****PART A:** 5 Questions of 6 Marks each – No choice**30 Marks****PART B:** 2 Questions from each unit of internal choice, each carrying 14 Marks**70 Marks**

SCSA6301	INTERNET OF THINGS LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

COURSE OBJECTIVES

- To identify the different technologies and different platforms in IoT
- To understand how to use sensors and actuators for design of IoT.
- To learn different protocols used in IOT.
- To learn the concepts of smart city development in IOT.
- To learn how to analysis the data in IOT.

SUGGESTED LIST OF EXPERIMENTS

1. Monitor the soil moisture level using Raspberry pi.
2. Monitor the surrounding air's quality using Arduino & sensors connected to a Raspberry Pi.
3. Implement Raspberry Pi as a local server by installing Apache, PHP, and MySQL.
4. Interface PIR sensor with ESP8266 WiFi module on NodeMCU platform.
5. Posting temperature and humidity of the environment to Thing speak cloud service using NodeMCU.
6. Implement Raspberry Pi Temperature System with Ubidots cloud service.
7. Implement smart cities Sensor Board v3.0 using Wasp mote platform for measuring temperature, humidity and pressure.
8. Measure light intensity and pollution level using Wasp mote platform.
9. Receiving and storing data in Meslium gateway for Wasp mote platform.
10. Implement barometric pressure monitoring using BMP280 sensor for Zolertia module.

SCS1101	PROBLEM SOLVING TECHNIQUES	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES:

- To understand the basics of solving a problem through computing
- To study different algorithms for solving a given problem.
- To understand and apply the algorithm analysis techniques.
- To critically analyze the efficiency of alternative algorithmic solutions for the same problem
- To understand different algorithm design techniques

COURSE OUTCOMES:

On completion of the course, student will be able to–

- Design Algorithms for various computing Problem.
- Analyze the time and space complexity of Algorithm.
- Examine the different algorithm design techniques for a given problem.
- Modify Existing algorithms to improve efficiency.
- Examine the different algorithm to solve the Mathematical problems.
- Apply the algorithm in real time scenario.

UNIT 1 PROGRAMMING TECHNIQUES**9 Hrs.**

Steps Involved in Computer Programming - Problem Definition - Outlining The Solution - Flow Chart - Developing Algorithms - Efficiency of Algorithms - Analysis of Algorithms - Step Count & Operation Count

UNIT 2 FUNDAMENTAL ALGORITHMS**9 Hrs.**

Exchanging the Values - Counting - Summation of Set of Number - Factorial Computation - Sine Computation - Fibonacci Sequence - Reversing the Digits of an Integer - Base Conversion - Character to Number Conversion.

UNIT 3 FACTORING METHODS**9 Hrs.**

Finding the Square Root of a Number - Smallest Divisor of an Integer - GCD of Two Integers - Generating Prime Numbers - Computing the Prime Factors of an Integer - Generation of Pseudo - Random Numbers - Raising a Number to a Large Power - Computing the Nth Fibonacci Number.

UNIT 4 ARRAY TECHNIQUES**9 Hrs.**

Array Order Reversal - Array Counting or Histogramming - Finding the Maximum Number in a Set - Removal of Duplicates from an Ordered Array - Partitioning an Array - Finding the kth Smallest Element - Longest Monotone Subsequence.

UNIT 5 DESIGN OF ALGORITHMS- BACKTRACKING, BRANCH AND BOUND, SEARCHING**9 Hrs.**

Backtracking - 8 Queens - Hamiltonian Circuit Problem, Branch and Bound - Travelling Salesman Problem, Searching - water jug problem, tic-tac-toe problem.

Max. 45 Hours

TEXT / REFERENCE BOOKS

1. Dromey.R.G, "How to Solve it by Computer", Prentice-Hall of India, Eighth Indian Reprint,1996.
2. Ellis Horowitz,sartaj Sahni ,"Fundamentals of Data Structures", Computer Science Press,1983.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****PART A :** 10 questions of 2 marks each- No choice**PART B :** 2 questions from each unit of internal choice, each carrying 16 marks**Exam Duration : 3 Hrs.****20 Marks****80 Marks**

SCS1103	DATA STRUCTURES	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES:

- To impart the basic concepts of data structures and algorithms
- To be familiar with writing recursive methods.
- To understand concepts about searching and sorting techniques.
- To understand basic concepts about stacks.
- To understand basic concepts about queues.
- To master the implementation of linked data structure such as linked lists.

COURSE OUTCOMES :

On completion of the course, student will be able to–

- Understand the concept of recursive algorithms
- Demonstrate the different types of data structures
- Able to understand the operations on linear data structures
- Ability to summarize searching and sorting techniques
- Ability to choose appropriate data structure as applied to specified problem definition
- Understand and implement the applications of linear data structures

UNIT 1 INTRODUCTION**9 Hrs.**

Introduction to algorithms - Recursion - Definition - Design Methodology and Implementation of recursive algorithms - Linear and binary recursion - recursive algorithms for factorial function - Fibonacci sequence - Tower of Hanoi - Tail recursion – Data Structures - Need - classification - operations - Array - characteristics - types - storage representations.

UNIT 2 SEARCHING AND SORTING TECHNIQUES**9 Hrs.**

Basic concepts - List Searches using Linear Search - Binary Search - Fibonacci Search - Sorting Techniques - Insertion sort - Heap sort - Bubble sort - Quick sort - Merge sort - Analysis of sorting techniques.

UNIT 3 STACKS**9 Hrs**

Basic Stack Operations - Representation of a Stack using Arrays - Algorithm for Stack Operations - Stack Applications: Reversing list - Factorial Calculation - Infix to postfix Transformation - Evaluating Arithmetic Expressions.

UNIT 4 QUEUES**9 Hrs**

Basic Queue Operations - Representation of a Queue using array - Implementation of Queue Operations using Stack - Applications of Queues - Round robin Algorithm - Enqueue - Dequeue - Circular Queues - Priority Queues.

UNIT 5 LINKED LISTS**9 Hrs**

Introduction - Single linked list - Representation of a linked list in memory - Operations on a singly linked list - Merging two singly linked lists into one list - Reversing a singly linked list - Applications of singly linked list to represent polynomial expressions and sparse matrix manipulation - Advantages and disadvantages of singly linked list - Circular linked list - Doubly linked list - Circular Doubly Linked List.

Max. 45 Hours

TEXT / REFERENCE BOOKS

1. Jean-Paul Tremblay, Paul G. Sorenson, 'An Introduction to Data Structures with Application', TMH, 2nd Edition.
2. Naps, Thomas L., and Bhagat Singh, "Introduction to Data Structure with Pascal", West Publishing Co., 1986.
3. Richard F, Gilberg, Forouzan, "Data Structures", Cengage, 2nd Edition.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max Marks : 80****Exam Duration : 3 Hrs.****PART A : 10 questions of 2 marks each- No choice****20 Marks****PART B : 2 questions from each unit of internal choice, each carrying 16 marks -****80 Marks**

SCS1201	ADVANCED DATA STRUCTURES	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES:

- To impart the concepts of data structures and algorithms
- To understand tree concepts.
- To solve problems on graphs.
- To implement routing algorithms.
- To describe the hash function and concepts of collision and its resolution methods and operations on set.

COURSE OUTCOMES:

On completion of the course, student will be able to

- Understand and implement the basics on non-linear data structures
- Impart the knowledge on Application of Balanced Trees
- Understand the basics of Graph and various terminologies
- Implement and apply the Traversal and Spanning Trees in real time applications
- Implement Single Source and All Pairs Shortest Path algorithms in routing problems
- Understand and implement Tables and Sets

UNIT 1 BASIC TREE CONCEPTS**9 Hrs.**

Trees- Ordinary and Binary trees terminology, Properties of Binary trees, Implementation using Array and Linked list - Binary tree ADT representations, recursive and non recursive traversals - Binary Search tree - Insertion and Deletion.

UNIT 2 ADVANCED TREE CONCEPTS**9 Hrs.**

Threaded Binary Trees, AVL Tree, B-tree Insertion and deletion, Splay trees - Heap trees - Heapify Procedure, Tries .

UNIT 3 GRAPH CONCEPTS**9 Hrs.**

Terminology, Representation using Array and Linked List - Types of graphs - Graph traversals - BFS and DFS - Applications.

UNIT 4 ADVANCED GRAPH CONCEPTS**9 Hrs.**

Minimum Spanning Tree - Kruskal's, Prim's and Sollin's Algorithm - Shortest path using Dijkstra's, Bellman Ford and Floyd Warshall Algorithm

UNIT 5 TABLES AND SETS**9 Hrs.**

Rectangular tables - Jagged tables - Inverted tables - Symbol tables - Static tree tables - Dynamic tree tables - Hash tables. Sets: Representation - Operations on sets - Applications.

Max. 45 Hours

TEXT / REFERENCE BOOKS

1. Ellis Horowitz and Sartaj Sahni "Fundamentals of Data Structures" Galgotia Book Source, Pvt. Ltd., 2004.
2. M. A. Weiss, "Data Structures and Algorithm Analysis in C", Second Edition, Pearson Education, 2005.
3. Naps, Thomas L., and Bhagat Singh. "Introduction to Data Structure with Pascal". West Publishing Co., 1986.
4. Jean Paul Tremblay and Paul G. Sorenson, "An Introduction to Data Structures with Applications", Tata McGraw-Hill, Second edition, 2001.
5. Aaron M Tanenbaum, Moshe J Augenstein and Yedidiah Langsam, "Data Structures using C and C++", Pearson Education, 2004.
6. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, "Data Structures and Algorithms", Pearson Education, First Edition Reprint 2003.
7. R. F. Gilberg, B. A. Forouzan, "Data Structures", Second Edition, Thomson India Edition, 2005.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A : 10 questions of 2 marks each- No choice****20 Marks****PART B : 2 questions from each unit of internal choice, each carrying 16 marks****80 Marks**

SCS1202	OBJECT ORIENTED PROGRAMMING	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES :

- To impart the knowledge to develop solutions to problems demonstrating usage of control structures, modularity, I/O.
- 2.To discuss the usage of data abstraction, encapsulation, and inheritance.
- 3.To understand the difference between object-oriented programming and procedural programming.
- 4.To implement object-oriented programming to develop solutions to problems demonstrating usage of control structures, modularity, I/O.
- 5.To analyze the problem-solving skills with appropriate encapsulation and design principles.

COURSE OUTCOMES :

On completion of the course, student will be able to–

- To understand how C++ improves C with object-oriented features.
- To understand the concept of encapsulation and inheritance and polymorphism.
- To analyze how to design C++ classes for code reuse.
- To implement copy constructors, class member functions ,inline functions and friend functions.
- To study how inheritance and virtual functions implement dynamic binding with polymorphism.
- To learn how to design and implement generic classes with C++ templates

UNIT 1 INTRODUCTION TO OBJECT ORIENTED PROGRAMMING**9 Hrs.**

Object Oriented Programming Paradigms - Comparison of Programming Paradigms - Basic Object Oriented Programming concepts - Comparison with C - Overview of C++ - Pointers - Functions - Scope and Namespaces - Source files and programs.

UNIT 2 CLASSES AND OBJECTS**9 Hrs.**

Working with classes - Classes and objects - Class specification - Defining class members - Objects Accessing member functions - Inline Functions - Data hiding - Class member accessibility - Empty classes.

UNIT 3 CONSTRUCTORS AND OVERLOADING**9 Hrs.**

Default constructors - Parameterized constructors - Constructor overloading - Copy constructors - new, delete operators-"this" pointer - friend classes and friend functions - Function overloading- Unary Operator overloading - Binary Operator overloading.

UNIT 4 INHERITANCE**9 Hrs.**

Base class and derived class relationship - Derived class declaration - Forms of inheritance - Inheritance and member accessibility - Constructors in derived class - Destructors in derived class - Multiple inheritance - Multi level inheritance - Hybrid inheritance - Virtual base classes - Member function overriding - Virtual functions - Abstract classes - Pure Virtual functions.

UNIT 5 I/O AND LIBRARY ORGANIZATION**9 Hrs.**

I/O Stream - File I/O - Exception Handling - Templates - STL - Library Organization and Containers - Standard Containers -Overview of Standard Algorithms - Iterators and Allocators.

TEXT / REFERENCE BOOKS

1. Balagurusamy, "Object Oriented Programming with C++", Tata McGraw Hill, 4th Edition, 2010.
2. Venu Gopal K.R, Ravishankar.T, and Raj kumar, "Mastering C++", Tata McGraw Hill, 1999.
3. Bjarne Stroustrup, "The C++ programming language", Addison Wesley, 3rd Edition, 1998.
4. John R Hubbard, "Programming with C++", Schaums Outline Series, McGraw Hill, 2nd edition, 2009.
5. James Martin & James J.Odell, "Object Oriented methods - A foundation", Prentice Hall, 1 997.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****PART A :** 10 questions of 2 marks each- No choice**PART B :** 2 questions from each unit of internal choice, each carrying 16 marks**Exam Duration : 3 Hrs.****20 Marks****80 Marks**

SCS1203	FUNDAMENTALS OF DIGITAL SYSTEMS	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES :

- To have a knowledge about the fundamentals of Digital Electronics.
- To understand basics of Logic Gates.
- To develop ability to understand Combinational Logic.
- To develop ability to understand Sequential Logic.
- Also to have a knowledge about Memory units.

COURSE OUTCOMES :

On completion of the course, student will be able to–

- Examine the structures for various number systems, their conversions and use of various logic gates with different input patterns.
- Understand the various methods to simplify Boolean Functions
- Explain the conceptual design of Programmable Logic Devices
- Demonstrate the functionality of various flip flops and the conversion between them.
- Use the flip flops to design and construct the various types of shift registers and Counters.
- Analyze and design different combinational, sequential logic circuits.

UNIT 1 NUMBER SYSTEMS, COMPLIMENTS AND CODES**9 Hrs.**

Number Systems - Binary Numbers - Number base conversions - Octal and Hexa Decimal Numbers - Complements - Signed Binary Numbers - Binary Arithmetic - Binary Codes - Decimal Code - Error Detection code - Gray Code- Reflection and Self Complementary codes - BCD number representation - Alphanumeric codes ASCII/EBCDIC - Hamming Code- Generation, Error Correction.

UNIT 2 BOOLEAN ALGEBRA AND LOGIC GATES**9 Hrs.**

Axiomatic definitions of Boolean Algebra - Basic Theorems and Properties of Boolean Algebra - Boolean Functions- Canonical and Standard forms - Digital Logic Gates- Simplification of Boolean Expressions: The map method- SOP and POS - NAND and NOR implementation - Don't Cares - The Tabulation Method - Determination and Selection of Prime Implicants

UNIT3 COMBINATIONAL LOGIC**9 Hrs.**

Design Procedure - Adder - Subtractor - Code Conversion - Analysis Procedure - Multilevel NAND/NOR circuits - Exclusive OR functions - Binary adder and subtractor- Decimal adder - BCD adder - Magnitude Comparator - Decoders - Demultiplexer - Encoder - Multiplexers

UNIT 4 SYNCHRONOUS SEQUENTIAL LOGIC**9 Hrs.**

Flip Flops - Analysis of clocked sequential circuit - Reduction and Assignments - Flip flop excitation tables - Design Procedure - Design of counters - Registers - Shift registers - Synchronous Counters - Timing sequences - Algorithmic State Machines - ASM chart - timing considerations - control implementation.

UNIT 5 ASYNCHRONOUS SEQUENTIAL LOGIC AND MEMORY**9 Hrs.**

Circuits with Latches - Analysis procedure and Design Procedure - Reduction of state and Flow tables - Race - Free State Assignment

TEXT / REFERENCE BOOKS

1. Morris Mano, "Digital Logic & Computer Design", Prentice Hall India, 2006.
- 2 Thomas L Floyd, "Digital Fundamentals", 10th Edition, Pearson Education, 2009.
- 3 Thomas.C.Bartee, "Computer Architecture & Logic Design", McGraw-Hill, 1991.
- 4 P.Malvino and D.P.Leach, "Digital Principles and Applications", 6th Edition, McGraw-Hill, 2006.
- 5 Thomas C. Bartee, "Computer Architecture Logic Design", 3rd Edition, 2002.

END SEMESTER EXAM QUESTION PAPER PATTERN

Max. Marks : 100

PART A : 10 questions of 2 marks each- No choice

PART B : 2 questions from each unit of internal choice, each carrying 16 marks

Exam Duration : 3 Hrs.

20 Marks

80 Marks

SCS1204	THEORY OF COMPUTATION	L	T	P	Credits	Total Marks
		3	1	0	4	100

COURSE OBJECTIVES :

- To have an understanding of finite state automata and pushdown automata.
- To understand regular languages and context free languages.
- To study the Turing machine and classes of problems.

UNIT 1 FINITE AUTOMATA AND REGULAR LANGUAGES**15 Hrs.**

Finite automata and regular languages - Regular languages and regular expressions - Finite automata - Union, intersections, and complements of automata - Non-determinism and Kleene's theorem - Non-deterministic finite automata and NFA with transition - Pumping lemma for Regular Languages

UNIT 2 CONTEXT-FREE LANGUAGES AND NORMAL FORMS**12 Hrs.**

Context-free grammars - Definition - More examples - Including some familiar languages - Union, concatenations, and *s of CFLs - Derivation trees and ambiguity - Unambiguous CFG for algebraic expressions - Normal Forms - CNF - GNF- Pumping Lemma for CFL.

UNIT 3 PUSH DOWN AUTOMATA**12 Hrs.**

Pushdown automata - Introduction - Definition - Deterministic pushdown automata - PDA corresponding to a given context-free grammar – Context-free Grammar corresponding to PDA.

UNIT 4 TURING MACHINES**14 Hrs.**

Turing machines - Models of computation and the Turing thesis - Definition of TM and TM as language acceptor - Non-deterministic TM and Deterministic TM – Universal TM

UNIT 5 RECURSIVE LANGUAGE AND UNSOLVABLE PROBLEMS**7 Hrs.**

Recursively enumerable and recursive languages - Recursively enumerable and recursive sets - Enumerating a language - Not all languages are recursively enumerable.

Unsolvability problem - An unsolvable decision problem - Reducing one problem to another (halting problem) - Other unsolvable problems in TM - Rice's theorem and more unsolvable problems.

Max. 60 Hours**TEXT / REFERENCE BOOKS**

1. John C. Martin, "Theory of Computation", 3rd Edition. Tata McGraw Hill, 2003.
2. Bernard M. Moret, "The Theory of Computation", Addison Wesley, 2002 Reprint.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A :** 10 questions of 2 marks each- No choice**20 Marks****PART B :** 2 questions from each unit of internal choice, each carrying 16 marks**80 Marks**

SCS1205	OBJECT ORIENTED ANALYSIS AND DESIGN	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES :

- To identify use cases and develop the use case model
- To develop the business activities and identify an UML activity diagram
- To identify the conceptual classes and develop a domain model with UML class diagram
- Using the identified scenarios find the interaction between objects and represent them using UML interaction diagrams
- To draw state chart diagrams component and deployment diagrams for real-time applications
- To improve the design by applying appropriate design patterns

COURSE OUTCOMES :

On completion of the course, student will be able to–

- Knowledge about the Object oriented concepts and different methodologies
- Can be able to use an object-oriented method for analysis and design
- Understand how to use UML for notation
- Ability to analyze and model software specifications.
- Have a general understanding of Quality Assurance activities and Testing Process
- Knowledge about the Object oriented concepts and different methodologies

UNIT 1 AN OVERVIEW OF OBJECT ORIENTED SYSTEM DEVELOPMENT**9 Hrs.**

Introduction - Object Oriented System Development Methodology - Why Object Orientation - Overview of Unified Approach - Object Basics: Object Oriented Philosophy - Objects - Classes - Attributes - Object Behavior and Methods, Encapsulation and Information Hiding - Class Hierarchy - Polymorphism - Object Relationships and Associations - Aggregations and Object Containment - Object Identity - Static and Dynamic Binding - Persistence. Object Oriented Systems Development Life Cycle: Software Development Process - Building High Quality Software - Usecase Driven Approach - Reusability.

UNIT 2 OBJECT ORIENTED METHODOLOGIES**9 Hrs**

Rumbaugh et al.'s Object Modeling Technique - Booch Methodology - Jacobson et al. Methodologies - Patterns - Framework - Unified approach - Unified Modeling Language: Static and Dynamic Model - UML Diagrams - UML Class Diagram - UML Usecase Diagram - UML Dynamic Modeling - UML Extensibility - UML Metamodel.

UNIT 3 OBJECT ORIENTED ANALYSIS**9 Hrs.**

Business Object Analysis - Use Case Driven Object Oriented Analysis - Business Process Modeling - Use Case model - Developing Effective Documentation - Object Analysis Classification: Classification Theory - Noun Phrase Approach - Common Class Patterns Approach - Use-Case Driven Approach - Classes Responsibilities and Collaborators - Naming Classes - Identifying Object Relationships, Attributes and Methods: Association - Super-Subclass Relationships - A-part of Relationships.

UNIT 4 OBJECT ORIENTED DESIGN**9 Hrs.**

Object Oriented Design Process - Object Oriented Design Axioms - Corollaries - Designing Classes: Object Constraint Language - Process of Designing Class - Class Visibility - Refining Attributes - Access Layer: Object Store and Persistence - Database Management System - Logical and Physical Database Organization and Access Control -

Distributed Databases and Client Server Computing - Object Oriented Database Management System - Object Relational Systems - Designing Access Layer Classes - View Layer: Designing View Layer Classes - Macro Level Process - Micro Level Process - Purpose of View Layer Interface - Prototyping the user interface.

UNIT 5 SOFTWARE QUALITY**9 Hrs.**

Software Quality Assurance: Quality Assurance Test - Testing Strategies - Impact of Object Oriented Testing - Test Cases - Test Plan - Myers Debugging Principle - System Usability and Measuring User Satisfaction: Usability Testing - User Satisfaction Testing

Max. 45 Hours**TEXT / REFERENCE BOOKS**

1. Ali Bahrami, "Object oriented systems development using the unified modeling language", 1st Edition, McGraw- Hill, 1998.
- 2 Grady Booch, James Rumbaugh, and Ivar Jacobson, "The Unified Modeling Language User Guide", 3rd Edition Addison Wesley, 2007.
- 3 John Deacon, "Object Oriented Analysis and Design", 1st Edition, Addison Wesley, 2005.
- 4 Bernd Oestereich, "Developing Software with UML, Object - Oriented Analysis and Design in Practice", Addison-Wesley, 2000.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A :** 10 questions of 2 marks each- No choice**20 Marks****PART B :** 2 questions from each unit of internal choice, each carrying 16 marks.**80 Marks**

SCS1206	DESIGN AND ANALYSIS OF ALGORITHM	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES :

- To analyze the asymptotic performance of algorithms.
- To learn mathematical background for analysis of algorithm.
- To learn various algorithm design techniques.
- To analyze the efficiency of the algorithm.
- To understand Np-Hard, NP-Complete problems.

COURSE OUTCOMES:

On completion of the course, student will be able to–

- Determine the suitable algorithmic design technique for a given problem.
- Identify the limitations of algorithms in problem solving
- Analyze the efficiency of the algorithm based on time and space complexity.
- Implement asymptotic notations to analyze worst-case and average case running times of algorithms.
- Interpret the fundamental needs of algorithms in problem solving.
- Describe various algorithmic techniques and its real time applications

UNIT 1 INTRODUCTION**9 Hrs.**

Fundamentals of Algorithmic Problem Solving - Time Complexity - Space complexity with examples - Growth of Functions - Asymptotic Notations: Big Oh, Little Oh, Omega, Theta - Properties - Complexity Analysis Examples - Performance measurement - Instance Size, Test Data, Experiment setup.

UNIT 2 MATHEMATICAL FOUNDATIONS**9 Hrs.**

Solving Recurrence Equations - Substitution Method - Recursion Tree Method - Master Method - Best Case - Worst Case - Average Case Analysis - Sorting in Linear Time - Lower bounds for Sorting - Counting Sort - Radix Sort - Bucket Sort

UNIT 3 DESIGN OF ALGORITHMS - BRUTE FORCE AND DIVIDE-AND-CONQUER**9 Hrs.**

Brute Force - Travelling Salesman Problem - Knapsack Problem - Assignment Problem - Closest Pair and Convex Hull Problems - Divide and Conquer Approach - Binary Search - Quick Sort - Merge Sort - Strassen's Matrix Multiplication.

UNIT 4 DESIGN OF ALGORITHMS - DYNAMIC PROGRAMMING AND GREEDY APPROACH**9 Hrs.**

Dynamic Programming - Floyd Warshall Algorithm - Optimal Binary Search Algorithms - Greedy Approach - Huffman Code - Kruskal's Algorithm - Prim's Algorithm - Dijkstra's Algorithm

UNIT 5 DESIGN OF ALGORITHMS - BACKTRACKING AND BRANCH AND BOUND**9 Hrs.**

Backtracking - 8 Queens - Hamiltonian Circuit Problem - Branch and Bound - Assignment Problem - Knapsack Problem - Travelling Salesman Problem - NP Complete Problems - Clique Problem - Vertex Cover Problem

Max. 45 Hours

TEXT / REFERENCE BOOKS

1. Sartaj Sahni, "Data Structures, Algorithms, and Applications in C++", McGraw Hill, 2nd Edition, 2005.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", Third Edition, PHI Learning Private Limited, 2012.
3. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", Third Edition, Pearson Education, 2012.
4. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, "Data Structures and Algorithms", Pearson Education, Reprint 2006.
5. Donald E. Knuth, "The Art of Computer Programming", Volumes 1 & 3 Pearson Education, 2009.
6. Steven S. Skiena, "The Algorithm Design Manual", Second Edition, Springer, 2008.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A :** 10 questions of 2 marks each - No choice**20 Marks****PART B :** 2 questions from each unit of internal choice, each carrying 16 marks**80 Marks**

SCS1207	AND ORGANISATION	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES :

- To impart knowledge on various types of Registers and Microinstructions.
- To discuss about the Data path design and Processor design in detail.
- To understand the types of memory organizations.
- To discuss about the registers, interface and communication in I/O devices.
- To analyze about the characteristics, structure, communication and synchronization of multiprocessor.

COURSE OUTCOMES :

On completion of the course, student will be able to–

- Classify various types registers, microinstructions and addressing modes.
- Design the data path for processor
- Demonstrate the usage Memory Organization.
- Describe about the I/O devices.
- Explain the characteristics, structure, communication and synchronization of multiprocessor.
- State the interprocessor communication and synchronization

UNIT 1 INTRODUCTION**10 Hrs.**

Central Processing unit - Introduction - General Register Organization - Stack organization - Basic computer Organization - Instruction codes - Computer Registers - Computer Instructions - Instruction Cycle - Arithmetic – Logic - Shift Microoperations - Arithmetic Logic Shift unit - Example Architectures: MIPS – Power – PC – RISC - CISC

UNIT 2 DATA PATH DESIGN**10 Hrs.**

Computer arithmetic : Addition - Subtraction - Multiplication and Division algorithms - Floating Point Arithmetic operations
Microprogrammed Control : Control memory - address sequencing - Microprogram Example - Design of Control unit - Example Processor design

UNIT 3 MEMORY ORGANISATION**8 Hrs.**

Memory Organization : Memory Hierarchy - Main memory - auxiliary Memory - Associative Memory - Cache Memory - Virtual memory

UNIT 4 IO ORGANISATION**9 Hrs**

Input - Output Organization : Peripheral Devices - I/O Interface - Modes of transfer - Priority Interrupt - DMA - IOP - Serial Communication

UNIT 5 MULTIPROCESSORS**8 Hrs.**

Characteristics of multiprocessors - Interconnection Structures - Interprocessor Arbitration - Interprocessor Communication and Synchronization - Cache coherence

Max. 45 Hours

TEXT / REFERENCE BOOKS

1. M.Morris Mano, 'Computer system Architecture",Prentice-Hall Publishers,Third Edition.
- 2 John P Hayes , 'Computer architecture and Organization', McGraw Hill international edition, Third Edition.
- 3 Kai Hwang and Faye A Briggs , 'Computer Architecture and Parallel Processing', McGraw Hill international edition,1995.

END SEMESTER EXAM QUESTION PAPER PATTERN

Max. Marks : 100

Exam Duration : 3 Hrs.

PART A : 10 questions of 2 marks each- No choice

20 Marks

PART B : 2 questions from each unit of internal choice, each carrying 16 marks

80 Marks

SCS1301	OPERATING SYSTEM	L	T	P	Credits	Total Marks
		3	0	0	3	100

UNIT 1 INTRODUCTION**8 Hrs.**

Introduction - Operating system structures - System components - OS services - System calls - System structure - Resources Processes - Threads - Objects - Device management - Different approaches - Buffering device drivers.

UNIT 2 PROCESS MANAGEMENT**9 Hrs.**

Processes - Process concepts - Process scheduling - Operations on processes - Cooperating processes - CPU scheduling - Basic concepts - Scheduling criteria - Scheduling algorithms - Preemptive strategies - Non-preemptive strategies

UNIT 3 SYNCHRONIZATION AND DEADLOCKS**9Hrs.**

The critical section problem - Semaphores - Classic problems of synchronization - Critical regions - Monitors - Dead locks - Deadlock characterization - Prevention - Avoidance - Detection - Recovery.

UNIT 4 MEMORY MANAGEMENT**9 Hrs.**

Storage Management Strategies - Contiguous Vs. Non-Contiguous Storage Allocation - Fixed & Variable Partition Multiprogramming - Paging - Segmentation - Paging/Segmentation Systems - Page Replacement Strategies - Demand & Anticipatory Paging - File Concept - Access Methods - Directory Structure - File Sharing - Protection - File - System Structure - Implementation.

UNIT 5 I/O SYSTEMS, LINUX & SHELL PROGRAMMING**10 Hrs.**

Mass Storage Structure - Disk Structure- Disk Scheduling - Disk Management - Swap Space Management - RAID Structure - Shell Operation Commands - Linux File Structure - File Management Operation - Internet Service - Telnet - FTP - Filters & Regular Expressions - Shell Programming - Variable, Arithmetic Operations, Control Structures, Handling Date, Time & System Information.

Max. 45 Hours.**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 : Comprehend knowledge about operating system components and services.
- CO2 : Apply knowledge of process scheduling algorithms for a given context.
- CO3 : Analyze process synchronization and deadlock conditions.
- CO4 : Construct the process of Mapping logical address to physical address.
- CO5 : Design appropriate strategies for Paging, Segmentation.
- CO6 : Develop real time applications Based on Linux shell programming.

TEXT / REFERENCE BOOKS

1. Abraham Silberschatz, Peter Galvin and Gagne, "Operating System Concepts", 6th Edition, Addison Wesley, 2002.
2. Harvey M.Deitel, "Operating System", 2nd Edition, Addison Wesley, 2000.
3. Gary Nutt, "Operating System, A modern perspective", 2nd Edition, Addison Wesley, 2000.
4. Richard Peterson, "Linux : The Complete Reference", 6th Edition, Tata McGraw Hills, 2007.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks: 100****Exam Duration: 3 Hrs.****PART A:** 2 Questions from each unit, each carrying 2 marks

20 Marks

PART B: 2 Questions from each unit with internal choice, each carrying 16 marks

80 Marks

SCS1302	COMPUTER GRAPHICS AND MULTIMEDIA SYSTEMS	L	T	P	Credits	Total Marks
		3	0	0	3	100

UNIT 1 BASICS OF COMPUTER GRAPHICS**9 Hrs.**

Output Primitives: Survey of computer graphics - Overview of graphics systems - Line drawing algorithm - Circle drawing algorithm - Curve drawing algorithm - Attributes of output primitives - Anti-aliasing.

UNIT 2 2D TRANSFORMATIONS AND VIEWING**8Hrs.**

Basic two dimensional transformations - Other transformations - 2D and 3D viewing - Line clipping - Polygon clipping - Logical classification - Input functions - Interactive picture construction techniques.

UNIT 3 3D CONCEPTS AND CURVES**10 Hrs.**

3D object representation methods - B-REP, sweep representations, Three dimensional transformations. Curve generation - cubic splines, Bezier's, blending of curves- other interpolation techniques, Displaying Curves and Surfaces, Shape description requirement, parametric function. Three dimensional concepts - Introduction - Fractals and self-similarity- Successive refinement of curves, Koch curve and peano curves.

UNIT 4 METHODS AND MODELS**8Hrs.**

Visible surface detection methods - Illumination models - Halftone patterns - Dithering techniques - Polygon rendering methods - Ray tracing methods - Color models and color applications.

UNIT 5 MULTIMEDIA BASICS AND TOOLS**10 Hrs.**

Introduction to multimedia - Compression & Decompression - Data & File Format standards - Digital voice and audio - Video image and animation. Introduction to Photoshop - Workplace - Tools - Navigating window - Importing and exporting images - Operations on Images - resize, crop, and rotate - Introduction to Flash - Elements of flash document - Drawing tools - Flash animations - Importing and exporting - Adding sounds - Publishing flash movies - Basic action scripts - GoTo, Play, Stop, Tell Target.

Max. 45 Hours**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 : Construct lines and circles for the given input.
- CO2 : Apply 2D transformation techniques to transform the shapes to fit them as per the picture definition..
- CO3 : Construct splines, curves and perform 3D transformations
- CO4 : Apply color and transformation techniques for various applications.
- CO5 : Analyze the fundamentals of animation, virtual reality, and underlying technologies
- CO6 : Develop photo shop applications. .

TEXT / REFERENCE BOOKS

1. Donald Hearn, Pauline Baker M., "Computer Graphics", 2nd Edition, Prentice Hall, 1994.
2. Tay Vaughan, "Multimedia", 5th Edition, Tata McGraw Hill, 2001.
3. Ze-Nian Li, Mark S. Drew, "Fundamentals of Multimedia", Prentice Hall of India, 2004
4. D. McClelland, L.U.Fuller, "Photoshop CS2 Bible", Wiley Publishing, 2005.
5. Jjames D. Foley, Andries van Dam, Steven K Feiner, John F. Hughes, "Computer Graphics Principles and Practice, 2nd Edition in C, Audison Wesley, ISBN - 981 -235-974-5.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A** : 10 questions of 2 marks each- No choice

20 Marks

PART B : 2 questions from each unit of internal choice, each carrying 16 marks

80 Marks

SCS1303	COMPILER DESIGN	L	T	P	Credits	Total Marks
		3	0	0	3	100

UNIT 1 INTRODUCTION**9 Hrs.**

Components of system software - editor - debugger - linker - loader - assembler - case study.

UNIT 2 BASICS OF COMPILER**9 Hrs.**

Compiler - Structure Of Compiler - Phases - Representation Of Lexical Phase Using Regular Expression - Representation Of Regular Expression - Finite Automata to Design Lexical Phase - Minimized DFA Algorithm.

UNIT 3 PARSER**9 Hrs.**

Types Of Parser - Shift Reduce Parsing - Operator Precedence Parsing - Recursive Decent Parser - Non-Recursive Decent Parser.

UNIT 4 INTERMEDIATE CODE GENERATION**9 Hrs.**

Intermediate code generation for assignment statements - boolean statements - switch case statement - symbol table generation.

UNIT 5 OPTIMIZATION**9 Hrs.**

Optimization - issues related to optimization - loop optimization - peep hole optimization - three address code generation algorithm - examples.

Max. 45 Hours**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 : Model a finite automata for any given regular expression.
- CO2 : Analyze various Parsing methods.
- CO3 : Generate the intermediate code and symbol table.
- CO4 : Apply code optimization methods to improve efficiency of the code
- CO5 : Formulate the issues involved in code Generation Process.
- CO6 : Construct the target code for given source code.

TEXT / REFERENCE BOOKS

1. D M. Dhamdhere , "System Programming", 2nd Edition, Tata McGraw Hill Publishing, 1999.
2. Alfred V.Aho, Jeffery D.Ullman & Ravi Sethi, " Compiler Principles, Techniques & Tools", Addison-Wesley Publishing Company, 1986
3. Alfred V.Aho, Jeffery D.Ullman "Principles of Compiler Design", Narosa Publishing House, 15th reprint, 1996.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A** : 10 questions of 2 marks each- No choice

20 Marks

PART B : 2 questions from each unit of internal choice, each carrying 16 marks

80 Marks

SCS1305	SOFTWARE ENGINEERING	L	T	P	Credits	Total Marks
		3	0	0	3	100

UNIT 1 INTRODUCTION**9 Hrs**

S/W engineering paradigm - Life cycle models - Water fall - Incremental - Spiral - Evolutionary - Prototyping - Object oriented system engineering - Computer based system - Verification - Validation - Life cycle process - Development process - System engineering hierarchy - Introduction to CMM - Levels of CMM.

UNIT 2 SOFTWARE ENGINEERING PROCESS**9 Hrs.**

Functional And Non-Functional - User - System - Requirement Engineering Process - Feasibility Studies - Requirements - Elicitation - Validation and management - Fundamental of requirement analysis - Analysis principles-Software prototyping - Prototyping in the Software Process - Rapid Prototyping Techniques - User Interface Prototyping - Software Document Analysis and Modeling - Data - Functional and Behavioral Models - Structured Analysis and Data Dictionary.

UNIT 3 DESIGN PROCESS AND CONCEPTS**9 Hrs.**

Design process - Modular design - Design heuristic - Design model and document - Architectural design - Software architecture - Data design - Architecture data - Transform and transaction mapping - User interface design - User interface design principles.

UNIT 4 BASIC CONCEPTS OF SOFTWARE TESTING**9 Hrs.**

Levels - Software Testing Fundamentals - Types of s/w test - White box testing- Basis path testing - Black box testing - Control Structure testing- Regression testing - Testing in the large- S/W testing strategies - Strategic approach and issues - UNIT testing - Integration testing - Validation testing - System testing and debugging. Case studies - Writing black box and white box testing.

UNIT 5 COST ESTIMATION & MAINTENANCE**9 Hrs.**

Software cost estimation - COCOMO model - Quality management - Quality concepts- SQA - Software reviews - Formal technical reviews - Formal approaches of SQA and software reliability - Software maintenance - SCM - Need for SCM - Version control - Introduction to SCM process - Software configuration items. Re-Engineering - Software reengineering - Reverse engineering - Restructuring - Forward engineering.

Max. 45 Hours**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 : Select appropriate Software Development Lifecycle Model for a given requirement.
- CO2 : Comprehend various Functional and Non-Functional Requirements.
- CO3 : Create User Interface using Design concepts.
- CO4 : Generate Test cases for the developed Code.
- CO5 : Estimate Cost & Maintenance of the Project..
- CO6 : Generate an overall plan for a specific application

TEXT / REFERENCE BOOKS

1. Pressman, "Software Engineering and Application", 6th Edition, McGraw International Edition, 2005..
2. Shooman. M.C , "Software Engineering Design", McGraw International Edition, 1985. Compiler Principles, Techniques & Tools", Addison-Wesley Publishing Company, 1986
3. Richard Fairley, " Software Engineering - Design, Reliability And Management", McGraw International Edition, 1983.
4. Sommerville, "Software Engineering", 6th Edition, Pearson Education, 2000.
5. Pfleeger, "Software Engineering", 6th Edition, Prentice Hall of India, 2005
6. Ghezzi, "Software Engineering", 2nd Edition, Prentice Hall of India, 2005.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A** : 10 questions of 2 marks each- No choice**20 Marks****PART B** : 2 questions from each unit of internal choice, each carrying 16 marks**80 Marks**

SCS1306	DATABASE MANAGEMENT SYSTEM	L	T	P	Credits	Total Marks
		3	0	0	3	100

UNIT 1 INTRODUCTION TO DATABASES**9 Hrs.**

Databases and Databases users - Database system concepts and architecture - Data modeling using Entity Relationship (ER) Model. **Relational Model** - The Relational Data Model and Relational Database Constraints - The Relational Algebra and Relational Calculus.

UNIT 2 DATABASE DESIGN**9 Hrs.**

Overview of the QBE Language - Overview of the Hierarchical Data Model - Overview of the Network Data Model - **SQL-99**: Schema Definition, Constraints, Queries, and Views- Functional Dependencies and Normalization for Relational Databases.

UNIT 3 QUERY PROCESSING**9Hrs.**

Algorithms for Query Processing and Optimization - Introduction to Transaction Processing Concepts and Theory - Concurrency control techniques.

UNIT 4 RECOVERY AND SECURITY**9Hrs.**

Database Recovery Techniques - Database Security - Distributed databases and Client- Server Architecture

UNIT 5 OBJECT DATABASE**9Hrs.**

Concepts for Object Database - Enhanced Data Models for Advanced Application **Emerging Database Technologies and Application**

Max. 45 Hours**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 : Construct ER model for a given application..
- CO2 : Apply normalization on Relational Databases
- CO3 : Demonstrate query optimization and concurrent transaction processing
- CO4 : Analyse recovery techniques, security mechanism, fragmentation and load balancing of databases.
- CO5 : Design database using Object Oriented techniques...
- CO6 : Create database for a specified application

TEXT / REFERENCE BOOKS

1. Elmasri & Navathe - Fundamentals of Database Systems - 3rd Edition, 2nd Reprint, Addison Wesley, 2000
2. Abraham Silberschatz, Henry. F. Korth and S. Sudharshan "Database System Concepts", 4th Edition, 2002.
3. C. J. Date, "Introduction to database systems", Addison Wesley, 7th Edition, 2001.
4. Jan L. Harrington, "Object oriented database design", Harcourt India private limited

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****PART A** : 10 questions of 2 marks each- No choice**PART B** : 2 questions from each unit of internal choice, each carrying 16 marks**Exam Duration : 3 Hrs.**

20 Marks

80 Marks

Max. 60 Hours

SCS4101	PROGRAMMING IN C LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

COURSE OBJECTIVES:

- To provide complete knowledge of C language.
- To introduce the fundamental concepts of programming to the students
- To develop logics which will help to create programs, applications in C.
- By learning the basic programming constructs they can easily switch over to any other language in future
- To familiarize them with art of writing programs.

COURSE OUTCOMES:

On completion of the course, student will be able to–

- Illustrate the flowchart and design and algorithm for a given problem
- Develop basic data types and input/output functions to write C programs
- Elaborate user defined functions, arrays, string manipulations to solve problems
- Apply user defined data types including structures and unions to solve problems
- Analyze files concept to show input and output of files in C
- Implement sorting and searching algorithm for data structures

SUGGESTED LIST OF EXPERIMENTS

1. Program to understand the basic data types and input/output functions.
2. Program for Looping and decision statements.
3. Program on Functions.
4. Program on Arrays.
5. Program on String Manipulations
6. Program on Structures and Union.
7. Program on Pointers.
8. Program to demonstrate the Command Line Arguments.
9. Program using Dynamic memory allocation.
10. Program to implement the Random Access in Files.
11. Program to implement math function.
12. Program to Implement sorting algorithms
13. Program to Implement searching algorithms
14. Programs to solve some of the Engineering applications.

SCS4102	DATA STRUCTURES LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

COURSE OBJECTIVES:

- To implement linear and non-linear data structures.
- To understand the different operations of search trees.
- To implement graph traversal algorithms.
- To get familiarized to sorting algorithms.
- To implementation of linear search and binary Search.

COURSE OUTCOMES:

On completion of the course, student will be able to–

- Remembering the concept of data structures through ADT including List, Stack and Queues.
- Understand basic concepts about stacks, queues, lists, trees and graphs.
- Able to apply and implement various tree traversal algorithms and ensure their correctness.
- Ability to analyze algorithms and develop algorithms through step by step approach in solving problems with the help of fundamental data structures.
- Compare and contrast Array based and Link based applications of typical data structures such as Stacks and Queues.
- Design applications and justify use of specific linear data structures for various applications.

SUGGESTED LIST OF EXPERIMENTS

1. Program to insert and delete an element from an array.
2. Program to sort the elements using insertion sort.
3. Program to sort the elements using quick sort.
4. Program to sort the elements using merge sort.
5. Program to implement operations on a Singly linked list.
6. Program to implement operations on a doubly linked list.
7. Program to implement a Stack using an array.
8. Program to implement a Stack using a Linked list.
9. Program to implement Queue using an array.
10. Program to implement Queue using a Linked list.
11. Program to convert an infix expression to postfix expression.
12. Program to implement display elements of a queue according to their priority.

SCS4201	OBJECT ORIENTED PROGRAMMING LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

COURSE OBJECTIVES:

- To impart the knowledge to develop solutions to problems demonstrating usage of control structures, modularity, I/O.
- To discuss the usage of data abstraction, encapsulation, and inheritance.
- To understand the difference between object-oriented programming and procedural programming.
- To implement object-oriented programming to develop solutions to problems demonstrating usage of control structures, modularity, I/O.
- To analyze the problem-solving skills with appropriate encapsulation and design principles.

COURSE OUTCOMES:

On completion of the course, student will be able to–

- Describe the procedural and object-oriented paradigm with concepts of streams, classes, functions, data and objects.
- Be able to understand the difference between object-oriented programming and procedural oriented language and data types in C++.
- Understand dynamic memory management techniques using pointers, constructors, destructors, etc.
- Describe the concept of function overloading, operator overloading, virtual functions and polymorphism.
- Classify inheritance with the understanding of early and late binding, usage of exception handling, generic programming.
- Be able to program using C++ features such as composition of objects, Operator overloading, inheritance, Polymorphism etc.

SUGGESTED LIST OF EXPERIMENTS

1. Develop a C++ program to implement a class, object creation, member function invocation concept.
2. Develop a C++ program to implement the various constructors and destructor concept.
3. Develop a C++ program to implement a friend function, Inline function.
4. Develop a C++ program to implement an operator (Unary & Binary) overloading concept.
5. Develop a C++ program to implement a function overloading concept.
6. Develop a C++ program to implement a run time polymorphism.
7. Develop a C++ program to implement the following inheritance types.
 - a. Single
 - b. Multiple
 - c. Multilevel
 - d. Hierarchical

e. Hybrid

8. Develop a C++ program to implement an Abstract class concept.
9. Develop a C++ program to implement a Virtual function.
10. Develop a C++ program to find the number of characters in a file.
11. Develop a C++ program to handle the exceptions.

Case Study

1. Categorization of living beings as humans, animals, birds, insects, etc., using inheritance.
2. Develop user defined manipulator for the following named
 - a) Rupees for displaying Rs. and sets the precision to 2.
 - b) Dollar for displaying \$.

SCS4202	OBJECT ORIENTED ANALYSIS AND DESIGN LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

Course Outcomes:

On completion of the course, student will be able to–

- Knowledge about the Object oriented concepts and different methodologies
- Can be able to use an object-oriented method for analysis and design
- Understand how to use UML for notation
- Ability to analyze and model software specifications.
- Have a general understanding of Quality Assurance activities and Testing Process
- Knowledge about the Object oriented concepts and different methodologies

SUGGESTED LIST OF EXPERIMENTS

1. Study experiment – Object Oriented Concepts
2. Study of UML diagrams – aim and scope of diagrams.
3. Study of CASE tools – ArgoUML, Rational ROSE, Altova UML
4. Practicing on the methods of identifying classes
 - a. Understand – what is a class?
 - b. Structure of Class – relate UML representation to programming
 - c. Identifying classes for project – application of CRC method, Noun Phrase Approach.
5. Understanding relationships
 - a. Association
 - b. Realization
 - c. Generalization
 - d. Aggregation
6. Practicing Class diagrams
 - a. Designing Classes
 - i. refining attributes and designing methods.
 - ii. roles, multiplicity, qualifier.
 - b. components
7. Identifying Use cases – scenario based approach
8. Understanding usecases
 - a. identifying regular flow and alternative flow of events
 - b. identifying constraints, pre and post conditions
9. Activity diagrams – components, understanding difference between flow chart and activity diagram.

10. Interaction diagrams – sequence diagrams, collaboration diagrams.
11. Deployment Diagram – scenarios of application.
 - a. Student Information System
 - b. Online Ticket Reservation system
 - c. Employee Payroll system
 - d. Online Banking Application
 - e. ATM processing
 - f. Stock Maintenance
 - g. Library Management System

SCS1309	NETWORK SECURITY	L	T	P	Credits	Total Marks
		3	1	0	4	100

UNIT 1 NETWORK SECURITY DESIGN**12 Hrs.**

Hacking - Vulnerabilities - Design Issues - Human Issues - Implementation Issues - Threats - Reconnaissance Attacks - Access Attacks - Information Disclosure Attacks - Denial of Service Attacks - Threat Defense - Secure Communication - Network Security Best Practices - SAFE Campus Design.

UNIT 2 THREATS IN NETWORKS**12Hrs.**

Network Security Controls - Firewalls - Intrusion Detection Systems - Secure Email - Intruder - Intrusion detection system - Virus and related threats - Countermeasures - Firewalls design principles - Trusted systems

UNIT 3 SECURITY PRACTICE**12 Hrs.**

X.509 Authentication services - E-mail security - IP security - Web security- Network perimeter security - Secured router configuration - Firewall - Design principle - Trusted systems - VPN - IDS - IPS penetration testing - NAT - Implementation of cryptography and security

UNIT 4 NETWORK SECURITY PROTOCOLS**12Hrs.**

SSH - RADIUS - SSL - Kerberos - TLS - IPSec - Voice over IP - IPSEC - X.509 Authentication service - Electronic mail security S/MIME - Application security - SSL - PGP - SET.

UNIT 5 SECURITY SERVICES**12Hrs.**

Computer Forensics and Cyber Laws - Data Recovery - Security Policies and Procedures - Security Lifestyle Management - Security Audit - Managed Security Services

Max. 60 Hours**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 : Comprehend type of attacks and network security violations.
- CO2 : Apply ciphering techniques to secure data transfer
- CO3 : Analyse authentication techniques for different network scenarios..
- CO4 : Design Internet Protocol Security architecture to identify the vulnerability of the Internet systems..
- CO5 : Develop an intrusion detection system to find the attacks in networks.
- CO6 : Conduct a case study on recent threats and attacks.

TEXT / REFERENCE BOOKS

1. William Stallings, Cryptography and Network Security, 6th Edition, Pearson Education, March 2013.
2. Behrouz A. Forouzan, Debdeep Mukhopadhyay, Cryptography and Network Security, 2nd edition, McGraw Hill, 2012 .
3. William Stallings, "Network Security Essentials : Applications and Standards", Fourth Edition, Pearson Education.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A** : 10 questions of 2 marks each- No choice

20 Marks

PART B : 2 questions from each unit of internal choice, each carrying 16 marks

80 Marks

SCS1310	COMPUTER NETWORKS	L	T	P	Credits	Total Marks
		3	1	0	4	100

UNIT 1 FUNDAMENTALS & LINK LAYER**12Hrs.**

Building a network - Requirements - Layering and protocols - Internet Architecture - Network software - Performance –Framing - Error Detection - Flow control

UNIT 2 MEDIA ACCESS & INTERNETWORKING**12 Hrs**

Media access control - Ethernet (802.3) - Wireless LANs - 802.11 - Bluetooth - Bridging - Basic Internetworking – IP – CIDR – ARP – DHCP –ICMP

UNIT 3 MEDIA ACCESS & INTERNETWORKING**12 Hrs**

Routing – RIP – OSPF - Metrics - Global Internet – Areas – BGP - IPv6 - Multicast - Addresses - Multicast routing – DVMRP – PIM.

UNIT 4 TRANSPORT LAYER**12 Hrs.**

Overview of Transport layer - UDP - Reliable byte stream (TCP) - Connection management - Flow control - Retransmission - TCP Congestion control - Congestion avoidance- QoS - Application requirements.

UNIT 5 APPLICATION LAYER**12 Hrs.**

Traditional applications - Electronic Mail – SMTP - POP3 – IMAP- MIME - HTTP - DNS- SNMP.

Max. 60 Hours**COURSE OUTCOMES**

On completion of the course, student will be able to

CO1 : Comprehend the network architecture and various protocols in different networks

CO2 : Demonstrate the functions of MAC layers

CO3 : Analyze different routing algorithms in networks

CO4 : Analyze the functions of Transport layer and its protocols TCP and UDP

CO5 : Create the different types of networks required for specific applications..

CO6 : Develop a LAN network model applying sub- netting and routing mechanisms

TEXT / REFERENCE BOOKS

1. Larry L. Peterson, Bruce S. Davie, "Computer Networks: A Systems Approach", Fifth Edition, Morgan Kaufmann Publishers, 2011.
2. James F. Kurose, Keith W. Ross, "Computer Networking - A Top-Down Approach Featuring the Internet", Fifth Edition, Pearson Education, 2009.
3. Nader. F. Mir, "Computer and Communication Networks", Pearson Prentice Hall Publishers, 2010.
4. Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, "Computer Networks: An Open Source Approach", McGraw Hill Publisher, 2011.
5. Behrouz A. Forouzan, "Data communication and Networking", Fourth Edition, Tata McGraw Hill, 2011.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A** : 10 questions of 2 marks each- No choice

20 Marks

PART B : 2 questions from each unit of internal choice, each carrying 16 marks

80 Marks

SCS1401	RICH INTERNET APPLICATIONS	L	T	P	Credits	Total Marks
		3	0	0	3	100

UNIT 1 RICH INTERNET APPLICATION OVERVIEW**9 Hrs**

Introduction to Web2.0 - Key characteristics of Rich Internet Application - Current Rich Internet Application platforms - Rich Internet Application benefits - Rich Internet Application patterns and best practices - Rich Internet Application architecture - Restful Web Services with Nodes.

UNIT 2 INTRODUCTION TO EXT JS**9 Hrs.**

Functions - Objects and Classes - Javascript in Web Browsers - Getting started with Ext JS - Creating your first Ext JS Application - MVC Basics.

UNIT 3 BUILDING RIA WITH EXT JS**9 Hrs.**

Fundamental Classes - Event Handling - Panels and Grids - Layouts and Widgets - Working with data.

UNIT 4 INTRODUCTION TO GWT**9 Hrs**

Overview - Why Gwt - GWT compiler - Cross browser compatibility. Basic GWT - GWT modules - Build and deploy simple GWT application - GWT Widget Library.

UNIT 5 BUILDING RIA WITH GWT**9 Hrs.**

GWT Panels and Layouts - Event Handling - Internationalisation - Advanced GWT - RPC and AJAX - Writing a service implementation - Handling browser back button functionality - MVP Design pattern.

Max. 45 Hours**COURSE OUTCOMES**

On completion of the course, student will be able to

- CO1 : Explain Rich Internet Applications architecture and its benefits.
- CO2 : Classify the function, classes and object in Ext JS Application.
- CO3 : Apply event handling methods using Ext JS for any given application.
- CO4 : Build GWT application using Widget Library functions..
- CO5 : Develop applications using RPC and AJAX.
- CO6 : Create browser back button functionality using MVP Design pattern..

TEXT / REFERENCE BOOKS

1. Shreeraj Shah, "Web 2.0 Security: Defending Ajax, RIA, and SOA", Course Technology PTR, 2007.
2. Krishna Sankar, Susan A. Bouchard, "Enterprise Web 2.0 Fundamentals", Cisco Press, 2010.
3. David Flanagan, "JavaScript: The Definitive Guide", 6th Edition, O'Reilly Media, Inc, March 2011.
4. Alessio Malizia, Ext JS Documentation Site "Mobile 3D Graphics", Springer, 2006.
5. Colin Ramsay, Shea Frederick, Steve Cutter, Blades, "Learning Ext JS", Packt Publishing, 2008.
6. Jesus Garcia, "Ext JS in Action", Manning Publications, 2010.
7. Federico Kerek, "Essential GWT Building for the Web with Google Web Toolkit 2", Addison-Wesley Professional, 2010.

END SEMESTER EXAM QUESTION PAPER PATTERN**Max. Marks : 100****Exam Duration : 3 Hrs.****PART A** : 10 questions of 2 marks each- No choice

20 Marks

PART B : 2 questions from each unit of internal choice, each carrying 12 marks

80 Marks

SCS4301	OPERATING SYSTEM LAB	L	T	P	Credits	Total Marks
		3	0	0	3	100

SUGGESTED LIST OF EXPERIMENTS

1. Data Definition Language (DDL)
2. Data Manipulation Language (DML)
3. Data Control Language (DCL)
4. Constraints and built-in Functions
5. Joins and Group-by Commands
6. PL/SQL Program using functions
7. PL/SQL Program to create Triggers
8. Consider any application and design using.
 - a. Normalization
 - b. Data Flow Diagram
 - c. Entity-Relation Diagram
 - d. Data Dictionary
 - e. Table Structure

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1 : Apply Linux Shell program for networking problems.
 CO2 : Analyze different types of scheduling algorithms for concurrent processes.
 CO3 : Formulate critical section problems for process synchronization.
 CO4 : Investigate cause and effect related to process deadlock.
 CO5 : Decide appropriate memory allocation strategy.
 CO6 : Create a real time application based on operating system concepts.

SCS4302	RDBMS LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

SUGGESTED LIST OF EXPERIMENTS

1. Data Definition Language (DDL)
2. Data Manipulation Language (DML)
3. Data Control Language (DCL)
4. Constraints and built-in Functions
5. Joins and Group-by Commands
6. PL/SQL Program using functions
7. PL/SQL Program to create Triggers
8. Consider any application and design using.
 - a. Normalization
 - b. Data Flow Diagram
 - c. Entity-Relation Diagram
 - d. Data Dictionary
 - e. Table Structure

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1 : Comprehend the underlying concepts of relational database management system
- CO2 : Analyze various DDL/DML/DQL/DCL commands in SQL.
- CO3 : Implement integrity constraints on a database using a state-of-the-art DBMS
- CO4 : Design a normalized database using normalization techniques
- CO5 : Create PL/SQL programs including stored Procedures, stored functions, cursors, triggers.
- CO6 : Design a schema for any given real world problem

SCS4303	COMPILER LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

SUGGESTED LIST OF EXPERIMENTS

1. Implement Token Separation for a given expression using LEX.
2. Use LEX tool to implement Lexical Analyzer.
3. Use LEX and YACC to implement parser.
4. Use LEX and YACC tool to implement Desktop Calculator.
5. Construction of NFA from a regular expression.
6. Construction of DFA from a regular expression.
7. Implement Recursive Descent Parser algorithm.
8. Implement Shift Reduce Parser algorithm.
9. Implement the backend of the compiler to produce three address code.
10. Implement Symbol Table management
11. Construct a simple compiler

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1 : Analyse tokens for the given source code using Lex tool.
- CO2 : Develop a parser using Lex and Yacc tool.
- CO3 : Construct Finite automata for any given regular expression.
- CO4 : Develop the parser for checking syntactic error.
- CO5 : Generate symbol table and target code for the given input.
- CO6 : Design a compiler to convert given source code to target code.

SCS4304	NETWORKING LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

SUGGESTED LIST OF EXPERIMENTS

1. Creation of Date Server, and also print the client's address on the Server.
2. Creation of UDP Server
3. Creation of Chat Program
4. Calculation of Checksum for packet data and file.
5. Program to implement HTTP Protocol
6. Creation of Mail Client
7. Creation of Web Server
8. Creation of TELNET Protocol
9. Implement FTP using TCP
10. WiFi Simulation
11. WiTotal Simulation
12. Router Configuration

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1 : Comprehend the different aspects of client-server networks and User Datagram Protocol (UDP) server models.
- CO2 : Make use of HTTP protocol over the internet and examine the formation, transmission in the web servers..
- CO3 : Develop a web server with the open source software
- CO4 : Implement File Transfer Protocol to communicate with remote server.
- CO5 : Design access points for a location using wi-fi Simulation.
- CO6 : Create and configure Router to communicate one network to another network.

SCS4401	DISTRIBUTED COMPUTING LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

SUGGESTED LIST OF EXPERIMENTS

1. Create a web page using HTML frames, Tables, ordered and unordered list.
2. Develop client side form validation using JQuery.
3. Create following web control using Google Web Tool Kit
 - a. Checkbox
 - b. Radio Button
 - c. Basic Button
 - d. Hyperlink
4. Create program to retrieve the Customer information using JSON & PHP
5. Creating and consuming a web service using .NET Platform
6. Implementing Semantic Elements in HTML5
 - a. <article>
 - b. <figure>
 - c. <footer>
 - d. <header>
 - e. <nav>
7. Create a program to retrieve Customer informations using JSON & PHP
8. Perform BIOS troubleshooting for clearing I/O devices
9. Develop an application to fetch information from a database with AJAX
10. Implementing Semantic Elements in HTML5
Using <article>, <figure>, <footer>, <header> and <nav>

COURSE OUTCOMES

On completion of the course, student will be able to

- CO1 : Comprehend software components of distributed computing systems.
 CO2 : Design forms in web pages and validate it using jQuery.
 CO3 : Design applications using Ajax, PHP and CSS.
 CO4 : Develop a web application using Google web tool kit.
 CO5 : Design web pages incorporating Semantic Elements in HTML5.
 .CO6 : Creating application and consume a web service using .NET Platform.

SCSX1025	WIRELESS AND MOBILE NETWORKS (Common to CSE & IT)	L	T	P	Credits	Total Marks
		3	0	0	3	100

UNIT I INTRODUCTION**10 hrs.**

Medium access control – SDMA – FDMA – TDMA – CDMA – Telecommunication systems – GSM – DECT – TETRA & UMTS – Satellite systems – LEO – MEO – GEO – Handover.

UNIT II BROADCAST SYSTEM & WIRELESS LAN**10 hrs.**

Broadcast systems – Cyclic repetition of data – Digital audio broadcasting – Digital video broadcasting – Wireless LAN – Infrared Vs radio transmission – IEEE 802.11 – HYPER LAN – Bluetooth.

UNIT III MOBILE NETWORK LAYER**10 hrs.**

Mobile network layer – Mobile IP – Dynamic host configuration protocol – Adhoc networks – Routing – Destination sequence distance vector – Dynamic source routing – Hierarchical algorithms – Alternative metrics.

UNIT IV MOBILE TRANSPORT LAYER**10 hrs.**

Mobile transport layer – Traditional TCP – Indirect TCP – Snooping TCP – Mobile TCP – fast retransmit/fast recovery – Transmission/timeout freezing – Selective retransmission – Transaction oriented TCP.

UNIT V WAP & WLL**10 hrs.**

World Wide Web – Wireless Application Protocol (WAP) – WAP Architecture – WAP Protocols – VoIP service for mobile networks – Wireless Local loop (WLL).

REFERENCE BOOKS:

1. Jochen Schiller, "Mobile Communications", 2nd Edition, Pearson Education, 2008.
2. Yi-Bing Lin, Imrich Ehlantec, John Wiley & Sons, "Wireless and Mobile Network Architectures", Inc., 2008.

UNIVERSITY EXAM QUESTION PAPER PATTERN

Max Marks : 80

Exam Duration : 3 hrs.

Part A: 2 Questions from each unit, each carrying 2 marks

20 marks

Part B: 2 Questions from each unit with internal choice, each carrying 12 marks

60 marks

SCSX1026	CRYPTOGRAPHY & NETWORK SECURITY (Common to CSE & IT)	L	T	P	Credits	Total Marks
		3	0	0	3	100

UNIT I INTRODUCTION**10 hrs.**

Services – Mechanisms and attacks – The OSI security architecture – A model for network security – Classical encryption technique – Symmetric cipher model – Substitution technique – Rotar machines – Steganography.

UNIT II BLOCK CIPHERS AND THE DATA ENCRYPTION STANDARD (DES)**10 hrs.**

Simplified DES – Block Cipher principles – The Data Encryption Standard – The strength of DES – Confidentiality using symmetric encryption – Placement of encryption – Traffic confidentiality – Key distribution – Random number generation.

UNIT III PUBLIC KEY ENCRYPTION AND KEY MANAGEMENT**10 hrs.**

Introduction to number theory – Public key cryptography and RSA – Key management – Diffie-hellman key exchange.

UNIT IV AUTHENTICATION AND HASH FUNCTIONS**10 hrs.**

Authentication requirements – Authentication functions – Message authentication codes – Hash functions – Security of hash functions and MAC'S – MD5 (Message Digest Algorithm) – HMAC.

UNIT V NETWORK SECURITY**10 hrs.**

Network security and system Security – Electronic mail security – IP security – Web security – Intruders – Malicious S/Ws – Firewalls.

REFERENCE BOOKS:

1. William Stallings, "Cryptography and Network Security", 2nd Edition, Prentice Hall of India, New Delhi, 1999.
2. Bruce Schneier, "Applied Cryptography", 2nd Edition, John Wiley and Sons, 1996.
3. Douglas R. Stinson, "Cryptography- Theory and Practice", CRC Press, 1995.

UNIVERSITY EXAM QUESTION PAPER PATTERN

Max Marks : 80

Exam Duration : 3 hrs.

Part A: 2 Questions from each unit, each carrying 2 marks

20 marks

Part B: 2 Questions from each unit with internal choice, each carrying 12 marks

60 marks

SCSX1027	HARDWARE PERIPHERALS AND INTERFACING	L	T	P	Credits	Total Marks
		3	0	0	3	100

UNIT I CPU AND MEMORY**10 hrs.**

CPU Essentials – Processor Modes – Modern CPU Concepts – Architectural Performance Features – The Intel's CPU – CPU Over Clocking – Over Clocking Requirements – Over Clocking The System – Over Clocking The Intel Processors – Essential Memory Concepts – Memory Organizations – Memory Packages – Modules – Logical Memory Organizations – Memory Considerations – Memory Types – Memory Techniques – Selecting And Installing Memory

UNIT II MOTHERBOARDS**10 hrs.**

Active Motherboards – Sockets And Slots – Intel D850GB – Pentium4 Mother Board – Expansion Slots – Form Factor – Upgrading A Mother Board – Chipsets – North Bridge – South Bridge – CMOS – CMOS Optimization Tactics – Configuring The Standard CMOS Setup – Motherboard BIOS – POST – BIOS Features – BIOS And Boot Sequences – BIOS Shortcomings And Compatibility Issues – Power Supplies And Power Management – Concepts Of Switching Regulation – Potential Power Problems – Power Management.

UNIT III STORAGE DEVICES**10 hrs.**

The Floppy Drive – Magnetic Storage – Magnetic Recording Principles – Data And Disk Organization – Floppy Drive – Hard Drive – Data Organization And Hard Drive – Sector Layout – IDE Drive Standard And Features – Hard Drive Electronics – CD-ROM Drive – Construction – CDROM Electronics – DVD-ROM – DVD Media – DVD Drive And Decoder.

UNIT IV I/O PERIPHERALS**10 hrs.**

Parallel Port – Signals And Timing Diagram – IEEE1284 Modes – Asynchronous Communication - Serial Port Signals – Video Adapters – Graphic Accelerators – 3D Graphics Accelerator Issues – Directx – Mice – Modems – Keyboards – Sound Boards – Audio Bench Marks.

UNIT V BUS ARCHITECTURE**10 hrs.**

Buses – Industry Standard Architecture (ISA), Peripheral Component Interconnect (PCI) – Accelerated Graphics Port (AGP) – Plug-And-Play Devices – SCSI Concepts – USB Architecture.

REFERENCE BOOKS:

1. Stephen J.Bigelow, "Trouble Shooting, maintaining and Repairing PCs", Tata McGraw-Hill, New Delhi, 2001.
2. Craig Zacker & John Rourke, "The complete reference:PC hardware", Tata McGraw-Hill, New Delhi, 2001.
3. Mike Meyers, "Introduction to PC Hardware and Trouble shooting", Tata McGraw-Hill, New Delhi, 2003.
4. B.Govindarajulu, "IBM PC and Clones hardware trouble shooting and maintenance", Tata McGraw-Hill, New Delhi, 2002.

UNIVERSITY EXAM QUESTION PAPER PATTERN

Max Marks : 80

Exam Duration : 3 hrs.

Part A: 2 Questions from each unit, each carrying 2 marks

20 marks

Part B: 2 Questions from each unit with internal choice, each carrying 12 marks

60 marks

SCSX1028	DISTRIBUTED COMPUTING (Common to CSE & IT)	L	T	P	Credits	Total Marks
		3	0	0	3	100

UNIT I FUNDAMENTALS**10 hrs.**

What is distributed computing systems – Evolution of distributed computing systems – Distributed computing system models – What is distributed operating system – Issues in designing distributed operating systems.

Message passing – Features of a good message-passing system – Issues in IPC by message passing – Synchronization – Buffering – Multidatagram messages – Encoding and decoding of message data – Failure handling – Group communication.

UNIT II REMOTE PROCEDURE CALLS**10 hrs.**

RPC Models – Transparency of RPC – Implementing RPC mechanism – Stub generation – RPC messages – Marshaling arguments & results – Server Management – Parameter-passing semantics – Call semantics – Communication protocols for RPCs – Complicated RPCs – Client server binding – Security – Special types of RPCs – Light weight RPC.

UNIT III DISTRIBUTED SHARED MEMORY**10 hrs.**

General architecture of DSM systems – Design & implementation issues of DSM – Granularity – Structure of shared memory space – Consistency models – Replacement strategy – Thrashing – Hetrogenous DSM – Advantages of DSM.

UNIT IV SYNCHRONIZATION AND MANAGEMENT**10 hrs.**

Synchronization – Clock synchronization – Mutual exclusion – Election algorithms – Deadlocks.-Resource Management – Task assignment approach – Load balancing approach – Load sharing approach - Process Management – Process migration – Threads.

UNIT V DISTRIBUTED FILE SYSTEMS**10 hrs.**

Desirable features of a good distributed file systems – File models – File accessing models – File sharing semantics – File caching schemes – File replications – Fault tolerance – Atomic transaction.

REFERENCE BOOKS:

1. Pradeep K Sinha, "Distributed Operating Systems, Concepts & Design", Prentice Hall of India, 2009.
2. Andrew S.Tanenbaum, "Distributed Operating Systems", 1st Edition, Prentice Hall of India, 1995.
3. Andrew S.Tanenbaum, and Steen, Maarten van, "Distributed Systems", 2nd Edition, Prentice Hall of India, 2007.

UNIVERSITY EXAM QUESTION PAPER PATTERN

Max Marks : 80

Exam Duration : 3 hrs.

Part A: 2 Questions from each unit, each carrying 2 marks

20 marks

Part B: 2 Questions from each unit with internal choice, each carrying 12 marks

60 marks

SCSX1029	SOFT COMPUTING (Common to CSE, IT & BIOINFO)	L	T	P	Credits	Total Marks
		3	0	0	3	100

UNIT I NEURAL NETWORKS**10 hrs.**

Introduction to ANS – Adaline – Back propagation network – Hopfield network – Boltzman machine – Self organizing maps.

UNIT II FUZZY LOGIC**10 hrs.**

Fuzzy sets – Fuzzy rules and fuzzy reasoning – Fuzzy inference system – Mamdani fuzzy model – Sugeno fuzzy model – Tsukamoto fuzzy model.

UNIT III NEURO FUZZY**10 hrs.**

Adaptive Neuro Fuzzy Inference System – Coactive neuro-fuzzy modelling – Classification and regression trees – Data Clustering Algorithm – Rule based structure – Neuro - Fuzzy control I – Neuro -Fuzzy control II – Fuzzy decision making. .

UNIT IV GENETIC ALGORITHM**10 hrs.**

Introduction – Implementation of GA – Reproduction – Crossover – Mutation – Coding – Fitness scaling – Application of GA.

UNIT V ARTIFICIAL INTELLIGENCE**10 hrs.**

Introduction – Searching techniques – First order Logic – Forward reasoning – Backward reasoning – Semantic – Frames.

REFERENCE BOOKS:

1. James A.Freeman, David M. Skapura, "Neural Networks, Algorithms, Applications and Programming Techniques ", Pearson Education, 1991.
2. George.J.Klir, Bo Yuan , "Fuzzy Sets and Fuzzy Logic – Theory and Application", 1995.
3. J.S.R.Jang, C.T.Sun, E-Mizutani, "Neuro – Fuzzy and Soft Computing ", Prentice Hall of India, 1997.
4. David E. Goldberg, "Genetic Algorithms – In Search, Optimization and Machine Learning ", Pearson Education, 2005.
5. Stuart J.Russel, Peter Norvig , "Artificial Intelligence A Modern Approach ", 2nd Edition, Pearson Education, 2003.

UNIVERSITY EXAM QUESTION PAPER PATTERN

Max Marks : 80

Exam Duration : 3 hrs.

Part A: 2 Questions from each unit, each carrying 2 marks

20 marks

Part B: 2 Questions from each unit with internal choice, each carrying 12 marks

60 marks

SCSX4014	HARDWARE AND PERIPHERALS LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

List of Experiments

1. Perform the overlocking of a Pentium – Class system.
2. Perform the troubleshooting of clearing parity / ECC error in memory.
3. Perform BIOS troubleshooting for clearing Harddisk errors.
4. Perform BIOS troubleshooting for clearing I/O devices
5. Perform SCSI troubleshooting for clearing errors related with SCSI.
6. Perform Parallel Port troubleshooting for clearing errors based on Printer port.

SCSX4015	CASE TOOLS & TESTING LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

List of Experiments

1. Practicing the different types of case tools such as (Rational Rose & other open source) used for all the phases of the software development life cycle
2. Study of various tools in the Rational Rose and Rational RSA suite.
3. Thorough study of the problem – Identify project scope, objectives, Infrastructure.
4. Software requirements analysis, describe individual phases / modules of the project, identify deliverables.
5. Use work products – use case diagrams, activity diagrams, interaction diagrams and add interfaces to class diagrams
6. Data modeling
7. Source code generators
8. Software testing
 - (a) Study various testing tools such as WinRunner, Load Runner, Test Director, Rational suite and other opensource tools.
 - (b) Carry out performance testing in Loadrunner using ASM
 - (c) Perform experiments to do the following
 - I. Requirements testing
 - II. Usecase scenario testing
 - III. Documentation testing

Suggested list of Applications

1. Student Mark Analyzing system
2. Quiz system
3. Online Ticket Reservation system
4. Payroll system
5. Course Registration System
6. ATM systems
7. Stock Maintenance

S11XPROJ	PROJECT WORK & VIVA VOCE	L	T	P	Credits	Total Marks
		0	0	30	15	100

The objective of the project work is to make use of the knowledge gained by the student at various stages of the degree course. Students, will also be permitted to undertake industrial/consultancy project Work, outside the department, in industries/Research labs.

There shall be three assessments during the semester by a review committee. The student shall make three presentations on the progress made before the committee at various stages of the Project work. The Head of the Department shall constitute the review committee for each branch of study. The total marks obtained in the three reviews, shall be taken in to account. There will be a viva-voce examination at the end of the Project work, conducted by one internal examiner and one external examiner. The total marks secured will be the sum of marks secured in the Project reviews and Viva Voce Examination.

Each student is required to submit a Project report on the project assigned to him by the department. The report should be based on the information available in the literature or data obtained by the student by way of experiments conducted in the laboratory/industry.

SCSX4014	HARDWARE AND PERIPHERALS LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

List of Experiments

1. Perform the overlocking of a Pentium – Class system.
2. Perform the troubleshooting of clearing parity / ECC error in memory.
3. Perform BIOS troubleshooting for clearing Harddisk errors.
4. Perform BIOS troubleshooting for clearing I/O devices
5. Perform SCSI troubleshooting for clearing errors related with SCSI.
6. Perform Parallel Port troubleshooting for clearing errors based on Printer port.

SCSX4015	CASE TOOLS & TESTING LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

List of Experiments

1. Practicing the different types of case tools such as (Rational Rose & other open source) used for all the phases of the software development life cycle
2. Study of various tools in the Rational Rose and Rational RSA suite.
3. Thorough study of the problem – Identify project scope, objectives, Infrastructure.
4. Software requirements analysis, describe individual phases / modules of the project, identify deliverables.
5. Use work products – use case diagrams, activity diagrams, interaction diagrams and add interfaces to class diagrams
6. Data modeling
7. Source code generators
8. Software testing
 - (a) Study various testing tools such as WinRunner, Load Runner, Test Director, Rational suite and other opensource tools.
 - (b) Carry out performance testing in Loadrunner using ASM
 - (c) Perform experiments to do the following
 - I. Requirements testing
 - II. Usecase scenario testing
 - III. Documentation testing

Suggested list of Applications

1. Student Mark Analyzing system
2. Quiz system
3. Online Ticket Reservation system
4. Payroll system
5. Course Registration System
6. ATM systems
7. Stock Maintenance

S11XPROJ	PROJECT WORK & VIVA VOCE	L	T	P	Credits	Total Marks
		0	0	30	15	100

The objective of the project work is to make use of the knowledge gained by the student at various stages of the degree course. Students, will also be permitted to undertake industrial/consultancy project Work, outside the department, in industries/Research labs.

There shall be three assessments during the semester by a review committee. The student shall make three presentations on the progress made before the committee at various stages of the Project work. The Head of the Department shall constitute the review committee for each branch of study. The total marks obtained in the three reviews, shall be taken in to account. There will be a viva-voce examination at the end of the Project work, conducted by one internal examiner and one external examiner. The total marks secured will be the sum of marks secured in the Project reviews and Viva Voce Examination.

Each student is required to submit a Project report on the project assigned to him by the department. The report should be based on the information available in the literature or data obtained by the student by way of experiments conducted in the laboratory/industry.