Course: BSc. Computer Science (Complementary)

CSC1C01 – Computer Fundamentals

Semester: 1

Course Number: 1

Contact Hours: 2T+2L

Number of Credits: 2

Number of Contact Hours: 64

Course Evaluation: Internal – 15 Marks + External – 60 Marks

Aim of the Course:

To impart the students with fundamental principles and operations of various units of computer and to impart them with the basic skill in application packages.

Objectives of the Course:

· To learn the basics of computer hardware units and how they work together

· To acquire basic skill with office packages

Prerequisites: Background of the basic science at +2 level

Course Outline

UNIT I [7T+6L]

Number systems- Non-positional number systems and positional number systems (Binary, Octal and Hexadecimal), Converting from one number system to another- decimal to a new base, converting to decimal from another bases, converting from base other than ten to base other than ten, short cut method for converting from binary to octal, octal to binary, binary to hexadecimal and hexadecimal to binary, Computer Codes (BCD, EBCDIC, ASCII) error detecting and correcting codes, parity bit, Hamming Code, computer arithmetic ,importance of binary, binary addition and subtraction.

UNIT II [6T+7L]

Boolean Algebra and Logic circuits- fundamental concepts of Boolean Algebra, postulates, Principle of duality, theorems of Boolean Algebra, Boolean functions, minimization, complement, canonicals forms, conversion between canonical forms. Logic Gates- AND, OR, NOT, NAND, NOR, XOR and XNOR, logic circuits, converting expression to logic circuit,

universal NAND and NOR gates, Exclusive OR and equivalence functions, Design of Combinational circuits (Half Adder, Subtractor and Full Adder)

UNIT III [6T+7L]

Basic Computer Organization-Input Unit, Output Unit, Storage Unit (Direct, Sequential and Random Access), CPU organization, Control Unit (micro programmed and hardwired control), primary storage, memory hierarchy, storage locations and addresses, storage capacity, bit, byte, nibble, RAM, ROM, PROM and EPROM, cache memory, registers. Secondary storage devices (Magnetic tape, Hard disk and CD drive)

UNIT IV [7T+6L]

I/O devices - Input Devices-identification and its use, keyboard, pointing devices (mouse, touch pad and track ball), Video digitizer, remote control, joystick, magnetic stripes, scanner, digital camera, microphone, sensor, and MIDI instruments, Output Devices identification and its use, monitor, printer (laser, inkjet, dot-matrix), plotter, speaker, control devices (lights, buzzers, robotic arms, and motors)

UNIT V [6T+6L]

Planning a Computer program, purpose of program planning, algorithm, flowchart - symbols, sample flowcharts, advantages and limitations.

Text Books:

1. Pradeep K. Sinha and Priti Sinha, Computer Fundamentals, BPB

References:

1. Peter Nortorn, Introduction to Computer, TMH
2. Rajaraman, V, Fundamental of Computers, Prentice Hall India
3. B. Ram, Computer Fundamentals

Lab List

Word Processing

· Paragraph formatting

· Newspaper style Document

· Table creation

· Mail merge

· Page formatting and printing

Spreadsheet

· Worksheet entries, including formula

· Formatting cells

· Chart creation

· Functions

Presentation Software

· Creating presentation

· Animations

· Sound

· Inserting picture

Course Plan

Unit/hours (time required)	Topics to be taught (input)	Procedure (process) Student centric Method of teaching	Activity	Learning outcome (output)	Assessment
Unit I (13 hours)	Number Systems, Converting from one number system to another, short cut method for converting from binary to octal, octal to binary, binary to hexadecimal and hexadecimal to binary, Computer Codes, Computer Arithmetic	● Lecture ● Problem-Solving	● Discussion ● Extra Problems as Assignments	Understanding the Basics of ● Number Systems ● Computer Codes ● Binary Arithmetic	● Evaluating Problem-Solutions ● Q and A
Unit II (13 hours)	Boolean Algebra and	● Lecture ● Problem-	● Discussion ● Simplificatio	Understanding ● Simplificatio	● Evaluating

	Logic circuits- Logic Gates, universal gates, Exclusive OR and equivalence functions, Design of Combinational circuits.	Solving ● Illustrations	ns ● Designing Circuits	ns of Boolean Algebra ● Logic Gates ● Circuit Designing	Problem-Solutions ● Test
Unit III (13 hours)	Basic Computer Organization -Input Unit, Output Unit, Storage Unit, CPU organization, primary storage, memory hierarchy, storage locations and addresses, storage capacity, bit, byte, nibble, RAM, ROM, PROM and EPROM, cache memory, registers. Secondary storage devices	● Lecture ● Lab Sessions ● Illustrating Block Diagrams	● Discussion ● Drawing Block Diagrams	Understanding Basic Computer Organization	● MCQ ● Q and A
Unit IV (13 hours)	I/O devices - Input Devices- identification and its use, keyboard, pointing	● Lecture ● Seminar	● Discussion ● Identification of I/O devices	Understand the functions and working of various I/O devices	● Quiz

	devices (mouse, touch pad and track ball), Video digitizer, remote control, joystick, magnetic stripes, scanner, digital camera, microphone, sensor, and MIDI instruments, Output Devices identification and its use, monitor, printer (laser, inkjet, dot-matrix), plotter, speaker, control devices (lights, buzzers, robotic arms, and motors)				
Unit V (12 hours)	Planning a Computer program, purpose of program planning, algorithm, flowchart - symbols, sample flowcharts, advantages	• Lecture • Lab Sessions • Illustrations	• Discussion • Writing Algorithms • Draw Flowcharts	Understand the Basics of Programming	• Evaluating Algorithms and Flowcharts • Test

	and limitations.				

Faculty In-charge

Dr Jisha Jose Panackal

CSC2C02 – Fundamentals of System Software, Networks and DBMS

Semester: 2

Course Number: 2

Contact Hours: 2T+2L

Number of Credits: 2

Number of Contact Hours: 64

Course Evaluation: Internal – 15 Marks + External – 60 Marks

Aim of the Course:

To impart the students with the basic concepts of system software, Computer Networks and Database.

Objectives of the Course:

· To learn the basic concepts of various system software

· To learn the basics of Computer Networks

· To learn the basics of Databases

Prerequisites: Background of the basic science at +2 level

Course Outline

UNIT I [6 T+6L]

System software - classification of programming languages (Machine, assembly & High level), Characteristics and Comparison, language processors (Assembler, Interpreter and Compiler), Operating Systems- Functions, types of OS (batch, multiprogramming, time sharing, real time and distributed)

UNIT II [7 T+6L]

Computer networks- goals of networking, network topologies, types of networks (LAN, MAN and WAN), network model, OSI model- 7 layers, Internet Layer- 5 layers, Communication Media-Guided (Twisted Pair, Coaxial Cable and Fiber Optic) and Unguided (microwave, satellite).

UNIT III [6 T+7L]

Database Management Systems-definition, structure of Database, data models (Record based Data model, Network model: - Basic Components, Record types, data types, links, relationships, Hierarchical model and Relational model)

UNIT IV [6 T+7L]

Structured query language - Create, insert, select, update, delete, alter, drop commands

UNIT V [7 T+6L]

HTML-hypertext, hyper media, understanding basic HTML tools- HTML editor, web browser, General structure of HTML document, different types of elements-doc type, comment element, structural element, HTML tags and attributes: <html>, <body>, <head>, <title>, <h1>,... ,<h6>,
, <table>, <img>, <hr>, adding links, background image to the body, creating lists.

References:

1. P. K Sinha, Fundamentals of Computers
2. D. M Dhamdhere, Operating System: A concept based Approach
3. Behrouz A Forouzan, Data Communication & Networking, MC Graw Hill
4. Joel Sklar,Principles of Web Page Design, Vikas Publications

Lab List

HTML

1. Simple HTML document creation
2. HTML document with tables
3. HTML document with various lists
4. HTML document with links to different parts of the same
5. documents and to separate documents

MySQL

1. Table creation
2. Data insertion and deletion
3. Data retrieval
4. Alteration of tables

Course Plan

Unit/hours (time required)	**Topics to be taught (input)**	**Procedure (process)** **Student centric Method of teaching**	**Activity**	**Learning outcome (output)**	**Assessment**
Unit I (12 hours)	System software, Classification of	● Lecture ● Illustrations	● Discussion ● Comparative Study	Understanding the Basics of	● MCQ ● Q and A

	programming languages Characteristics and Comparison, language processors, Operating Systems- Functions, types of OS			• Software • Computer Languages • Pre-processors • OS	
Unit II (13 hours)	Computer networks- goals of networking, network topologies, types of networks, network model, OSI model- 7 layers, Internet Layer- 5 layers, Communication Media.	• Lecture • Illustrations	• Discussion • Drawing various topologies, Layers of Network Models	Understanding • Computer Networks • Network Models	• Test • Q and A
Unit III (13 hours)	Database Management Systems- definition, structure of Database, data models	• Lecture • Illustrations	• Discussion • Comparative Study	Understanding DBMS	• Q and A • Test
Unit IV (13 hours)	Structured query language - Create, insert, select, update, delete, alter, drop	• PPT • Lab Sessions	• Discussion • Hands-on Experience	Understand SQL using MySQL	• Verification of Lab Exercises • Q and A

	commands				
Unit V (13 hours)	HTML- hypertext, hyper media, understanding basic HTML tools- HTML editor, web browser, General structure of HTML document, different types of elements- doc type, comment element, structural element, HTML tags and attributes, adding links, background image to the body, creating lists.	• PPT • Lab Sessions	• Discussion • Hands-on Experience • Creative Learning	Understand the Basics of Web Designing using HTML	• Evaluating Web sites • MCQ

Faculty In-charge

Dr Jisha Jose Panackal

CSC3C03 – Problem Solving Using C

Semester: 3

Course Number: 3

Contact Hours: 3T+2L

Number of Credits: 2

Number of Contact Hours: 80 Hrs.

Course Evaluation: Internal – Internal – 15 Marks + External – 60 Marks

Aim of the Course:

To equip the students with the basic concepts of problem solving using computers.

Objectives of the Course:

· To learn the concepts of programming.

· To learn the C language

Prerequisites:

· Background of the basic science at +2 level

Course Outline

UNIT I [9 T+7L]

Introduction to C- Structure of C program, Character Set, Keywords, Identifiers, Data Types, Qualifiers, Variables, Declarations, Symbolic Constants, Expressions, Statements, Different Types of Operators (Arithmetic, Logical, Relational & Equality, Unary and Conditional), Operator Precedence and Associativity, Library Functions, Comments, I/O functions-(Formatted scanf() &printf(), getchar (), putchar (), getche(), gets(), puts())

UNIT II [9 T+7L]

Control Statements- Selection Statements (if, if-else, else if ladder, switch), iteration (while, do while, for), jumping (goto, break, continue), Nested Control Statements

UNIT III [10 T+6L]

Structured Data types - Arrays (One dimensional and Two Dimensional), Character and String Functions, Structure (Definition, Processing-period Operator), Union

UNIT IV [10 T+6L]

User defined Functions - Advantages, Definition, Accessing functions, formal and Actual Parameters, Recursion, Storage Classes- Automatic, External, Static and Register Variable, Argument Passing Mechanism

UNIT V [11T+6L]

Pointers and data files- Pointers, advantages, declaration, operations on pointers, pointers and one dimensional arrays, dynamic memory allocation. Data files (sequential), file handling functions (fopen(), fclose(), fputc(), fgetc(), fgets(), fputs(), fscanf(), fprintf()

Text Book:

1. E Balagurusamy, Programming in Ansi C, Tata McGraw Hill

References:

1. Byran Gotfried, Programming with C, Schaum Series
2. Kezningham & Ritchie,Programming in C
3. Yashvant Kanetkar, Let us C, BPB publications
4. Mullish Cooper, The spirit of C, Jasco books
5. Herbert Schildt, The Complete reference C, Tata McGraw Hill

Course Plan

Unit/hours (time required)	**Topics to be taught (input)**	**Procedure (process)** **Student centric Method of teaching**	**Activity**	**Learning outcome (output)**	**Assessment**
Unit I (16 hours)	Introduction to C- Structure of C program, Character Set, Keywords, Identifiers, Data Types, Qualifiers, Variables, Declarations, Symbolic Constants, Expressions, Statements,	• Lecture • Lab Sessions	• Discussion • Hands-on Learning	Understanding the Basics of • Programming using C • Tokens • Operators • I/O functions	• Evaluating Programs • Q and A

	Different Types of Operators, Operator Precedence and Associativity, Library Functions, Comments, I/O functions				
Unit II (16 hours)	Control Statements- Selection Statements (if, if-else, else if ladder, switch), iteration (while, do while, for), jumping, Nested Control Statements	• Lecture • Illustrations • Program Logic Sharing	• Discussion • Hands-on Learning	Understanding • Control Statements • Program Implementations	• Program Test • MCQ
Unit III (16 hours)	Structured Data types - Arrays, Character and String Functions, Structure, Union	• Lecture • Lab Sessions	• Discussion • Hands-on Learning	Understanding Basic Data Structures	• Evaluating Programs • Q and A • Test
Unit IV (16 hours)	User defined Functions - Advantages, Definition, Accessing functions, formal and Actual	• Lecture • Lab Sessions	• Discussion • Hands-on Learning • Comparative Study	Understand the functions and working of various programs	• Evaluating Programs • Quiz

	Parameters, Recursion, Storage Classes- Automatic, External, Static and Register Variable, Argument Passing Mechanism				
Unit V (16 hours)	Pointers and data files- Pointers, advantages, declaration, operations on pointers, pointers and one dimensional arrays, dynamic memory allocation. Data files, file handling functions	• Lecture • Lab Sessions • Illustrations	• Discussion • Writing Programs • Hands-on Learning	Understand the Basics of Dynamic Programming and use of Pointers	• Evaluating Programs • Q and A • Test

Faculty In-charge

Dr. Sr. Mini T V

CSC4C04 – Data Structure Using C

Semester: 4

Course Number: 4

Contact Hours per Week: 5(3T+2L)

Number of Credits:2

Number of Contact Hours: 80 hrs

Course Evaluation: Internal – 15 Marks + External – 60 Marks

Objectives of the Course:

- To introduce the concept of data structures
- To make the students aware of various data structures
- To equip the students implement fundamental data structures

Prerequisites:

- Knowledge in C Programming Language

Course Outline:

Unit I [11 T+6L]

Primitive Data types and Abstract Data Types(ADT) - Introduction to data structures – definition - characteristics of data structures - categories of data structures – algorithm - space complexity and time complexity of an algorithm (concept only).

Unit II [7 T+6L]

Arrays and Singly Linked Lists - 1D, 2D and Multi-dimensional arrays – operations on arrays - Sparse Matrix Representation

Unit III [9 T+7L]

Lists- Linked List- Definition –Creation- Operations, Basics of Doubly Linked List, Circular Linked List.

Unit IV [11 T+7L]

Stack and Queues – Definition and Operations on stack - Implementation of Stack using arrays and linked lists - Applications of Stacks - Polynomial Addition Queues – Definition, Implementations of queue using arrays and linked lists – basics of Circular queue, Dequeue - Applications of queues.

Unit V [10 T+7L]

Searching and Sorting: Searching: Linear search & Binary search. Sorting – Linear sort - Bubble sort - Selection sort - Insertion sort - Quick sort - Merge sort – Comparisons and implementations.

Text Books:

1. SeymourLipschutz,"Data Structures",TataMcGraw Hill Publishing Company Limited, Schaum"s Outlines, New Delhi.
2. YedidyanLangsam,MosheJ.Augenstein,and AaronM.Tenenbaum, "Data Structures Using C", Pearson Education., New Delhi.
3. Horowitz and Sahani, "FundamentalsofdataStructures",GalgotiaPublicationPvt.Ltd.,NewDelhi.

Reference Books:

1. Trembley, J.P. And Sorenson P.G., "An Introduction to Data Structures With Applications", Mcgraw- Hill International Student Edition, New York.
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Addison- Wesley, (An Imprint of Pearson Education), Mexico City

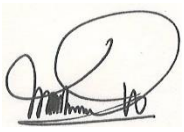
Course Plan

Unit/hours (time required)	**Topics to be taught (input)**	**Procedure (process) Student centric Method of teaching**	**Activity**	**Learning outcome (output)**	**Assessment**
Unit I (17 hours)	Primitive Data types and Abstract Data Types(ADT) - Introduction to data structures – definition - characteristics of data structures - categories of data structures – algorithm - space complexity and time	• Lecture • Lab Sessions	• Discussion • Hands-on Learning	Understanding the Basics of • Data Structures • Algorithms	• Q and A • Test

	complexity of an algorithm				
Unit II (13 hours)	Arrays and Singly Linked Lists - 1D, 2D and Multi-dimensional arrays – operations on arrays - Sparse Matrix Representation	• Lecture • Illustrations • Program Logic Sharing	• Discussion • Hands-on Learning	Understanding • Arrays • Matrices	• Program Test • MCQ
Unit III (16 hours)	Lists- Linked List- Definition –Creation- Operations, Basics of Doubly Linked List, Circular Linked List.	• Lecture • Lab Sessions	• Discussion • Hands-on Learning	Understanding Data Structures Using Linked Lists	• Evaluating Programs • Q and A • Test
Unit IV (18 hours)	Stack and Queues – Definition and Operations on stack - Implementation of Stack using arrays and linked lists - Applications of Stacks - Polynomial Addition Queues – Definition, Implementations of queue using arrays and linked lists – basics of Circular queue, Dequeue - Applications of queues.	• Lecture • Lab Sessions	• Discussion • Hands-on Learning • Comparative Study	Understand the functions of • Stacks • Queues And its implementations	• Evaluating Programs • Quiz
Unit V (17 hours)	Searching and Sorting:	• Lecture • Lab	• Discussion • Writing	Understand the Basics of	• Evaluating

	Searching: Linear search & Binary search. Sorting – Linear sort - Bubble sort - Selection sort - Insertion sort - Quick sort - Merge sort – Comparisons and implementations.	Sessions • Illustrations	Programs • Hands-on Learning	• Searching • Sorting	Programs • Q and A • Test

Faculty In-charge



Dr. Sr. Mini T V

Course: **MSc Computer Science (First Semester)**

Programme: CSS1C01 - Discrete Mathematical Structures

No. of lecture hours per week	4 Hrs
No.of credits for Theory	4

Aim of the course

To introduce discrete mathematics concepts necessary to understand basic foundation of Computer Science.

Course Outline

Unit I: Sets and Mathematical Logic: Set Theory - Types of sets, Set operations, Principles of Inclusion and Exclusion. Mathematical Logic - Propositional Calculus - Statement, Connectives, Conditional and Biconditional, Equivalence of Formula, Well Formed Formula, Tautologies, Duality Law, Functionally Complete Sets of Connectives, Normal Forms, Theory of Inference for the Statement Calculus, Predicate Calculus - Statement Functions, Variables and Quantifiers, Free and Bound Variables, Theory of Inference for the Predicate Calculus.

Unit II: Functions and Relations: Functions – Types of Functions, Composition of Functions and Inverse Functions. Relations - Relations and Their Properties, Functions as relations, Closure of Relations, Composition of relations, Equivalence Relations and Partitions. Partial Ordering, Hasse Diagram. The Pigeon Hole Principle.

Unit III: Lattices and Boolean Algebra - Lattices and Algebraic Systems, Principles of Duality, Basic Properties of Algebraic Systems Defined by Lattices, Distributive Lattices and Complemented Lattices. Boolean Lattices and Boolean Algebras. Boolean Functions and Boolean Expressions

.

Unit IV: Group Theory – Definition and Elementary Properties - Permutation Groups, Cyclic Groups – Subgroups - Cosets and Lagrange's Theorem, Semigroup and Monoid. Homeomorphism and Isomorphism. Rings, Integral Domains and Fields

.

Unit V: Graph Theory – Introduction, Directed Graph, Undirected Graph, Connected and Disconnected Graphs, Bipartite Graph, Complete Bipartite Graph, Isomorphic Graphs, Subgraph. Paths and Circuits. Shortest Paths in Weighted Graphs - Dijkstra's Algorithm. Eulerian Paths and Circuits, Hamiltonian Paths and Circuits. Trees - Spanning Trees and Cut-Sets, Minimum Spanning Trees - Kruskal's Algorithm, Prim's Algorithm.

References:

1. C Liu and D. Mohapatra, Elements of Discrete Mathematics - A Computer Oriented Approach, TMH, ISBN: 1259006395.
2. Alan Doerr and Kenneth Levassur, Applied Discrete Structure for Computer Science,

Galgotia Publications Pvt. Ltd, ISBN: 9780574217554.
3. J. K. Sharma, Discrete Mathematics, Macmillan Publishers India Limited, ISBN: 1403924759.
4. J. P. Tremblay and R. Manohar, Discrete Mathematical Structures with Application to Computer Science, McGraw-Hill Companies, ASIN: B001FPXR5Y.

Curriculum Plan

Unit/Hours (time required)	**Topics to be Taught (Input)**	**Procedure (process) student centric method of teaching**	**Activity**	**Learning outcome (outcome)**	**Assessment**
Unit1(18 Hour)	Sets and Mathematical Logic: Set Theory - Types of sets, Set operations, Principles of Inclusion and Exclusion. Mathematical Logic - Propositional Calculus - Statement, Connectives, Conditional and Biconditional, Equivalence of Formula, Well Formed Formula, Tautologies, Duality Law, Functionally Complete Sets of Connectives, Normal Forms, Theory of Inference for the Statement Calculus, Predicate Calculus - Statement Functions,	• Lecture using ICT tools	Peer group discussion on different types of sets	To Understand • Concept of different types of sets	• Assignment • MCQ evaluation

	Variables and Quantifiers, Free and Bound Variables, Theory of Inference for the Predicate Calculus.				
Unit 2 (20 Hours)	Functions and Relations: Functions – Types of Functions, Composition of Functions and Inverse Functions. Relations - Relations and Their Properties, Functions as relations, Closure of Relations, Composition of relations, Equivalence Relations and Partitions. Partial Ordering, Hasse Diagram. The Pigeon Hole Principle.	• Lecture using ICT Tools • Illustrations • Hands on sessions	• Seminar • Group stud	To Understand • Relations • Functions • Equivalence Relations • Partitions.	• Q&A exams.
Unit3 (18 Hours)	Lattices and Boolean Algebra - Lattices and Algebraic Systems, Principles of Duality, Basic Properties of Algebraic Systems Defined by Lattices, Distributive Lattices and Complemented Lattices. Boolean Lattices and	• Lecture using ICT tools • Illustrations	• Hands on sessions • Illustrations	To Understand Lattices and Boolean Algebra	• Q&A exams

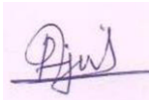
	Boolean Algebras. Boolean Functions and Boolean Expressions.				
Unit4 (14 Hours)	Group Theory – Definition and Elementary Properties - Permutation Groups, Cyclic Groups – Subgroups - Cosets and Lagrange's Theorem, Semigroup and Monoid. Homeomorphism and Isomorphism. Rings, Integral Domains and Fields.	• Lecture using ICT tools • Illustrations • Hands on Session	• Hands on sessions • Illustrations	To understand Group Theory	• Q&A exams
Unit5(20 Hours)	Graph Theory – Introduction, Directed Graph, Undirected Graph, Connected and Disconnected Graphs, Bipartite Graph, Complete Bipartite Graph, Isomorphic Graphs, Subgraph. Paths and Circuits. Shortest Paths in Weighted Graphs - Dijkstra's Algorithm. Eulerian Paths and Circuits, Hamiltonian Paths and Circuits. Trees - Spanning Trees and	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand Graph Theory	• Assignments • Exams

	Cut-Sets, Minimum Spanning Trees - Kruskal's Algorithm, Prim's Algorithm.				

Course Outcomes:

CO:1	CO1: Summarize different types of Sets
CO:2	CO2: Describe Inclusion Exclusion Principle
CO:3	CO3: Discuss Functions and Relations.
CO:4	CO4: Describe Equivalence relations and partitions.
CO:5	CO5: Discuss Lattice and Boolean algebra.
CO:6	CO6: Describe Boolean expressions and functions
CO:7	CO7: Discuss Integral Domain and Rings
CO:8	CO8: Describes Kruskal's Algorithm, Prim's Algorithm.

Faculty in Charge:



Julie P.A

Programme: CSS1C02 –ADVANCED DATA STRUCTURES

Number of Lecture hours per week: 3Hrs Number of practical hours:4Hrs

Number of credits for theory:5 Number of credits for practical:1

Aim of the course

To introduce basic and advanced data structures dealing with algorithm development and problem solving.

Course Outline

Unit I: Data structure - definition - types & operations, characteristics of data structures - Abstract Data Type (ADT) – algorithms - concepts - definition - bjectives of algorithms - quality of an algorithm - space complexity and time complexity of an algorithm.

Unit II: Counting Techniques: Basic counting techniques - permutations and combinations, asymptotic behaviour of functions. Linear data structures - Arrays records - representation - data structure operations - traversing, inserting and deleting - sorting and searching - sorting algorithms - linear search & binary search - complexity. Linked lists -

operations and implementations, - Stack - operations and its implementations (both array and linked list) - Applications - parsing arithmetic expressions, conversion and evaluating expressions. Recursion - characteristics of recursion, types of recursion applications of recursion in algorithms - comparison of recursive and non-recursive algorithms. Queue - operations and its implementations (both array and linked list) - circular queue - dequeue - priority queues, recursive lists, heterogeneous lists, deterministic skip lists, doubly linked lists and circular lists sparse matrix- representation.

Unit III: Non-linear Data Structures - trees - terminology - tree traversals algorithms Binary trees - threaded binary trees - binary search trees - traversals and operations on BST heap Tree - balanced trees - M-way trees - B and B+ trees, Red Black Tree, Digital Search Tree, Tries, Treaps, Huffman algorithm for extended binary tree - operations and their implementation. Graphs - representation of graphs – operations - traversals and their implementation.

Unit IV: Hashing - overview of hashing - hash tables - hash functions and their computations open addressing - linear probing - quadratic probing - double hashing algorithms and their implementations - rehashing - extendable hashing - separate chaining - hashing efficiency - heaps - overview of heaps - implementation and operations.

Unit V: Heap structures - Min-Max heaps - Deaps - leftist heaps - binomial heaps -Fibonacci heaps -binary heaps - skew heaps - pairing heaps - applications - amortized analysis an unrelated puzzle - Binomial queues - skew heaps - Fibonacci heaps - Splay trees.

References:

1. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, *Data Structures andAlgorithms*, Addison-Wesley, ISBN: 978-0201000238.
2. Horowitz E and Sahni S, *Fundamentals of Data Structures*, Computer Science Press, ISBN: 9780716780427.
3. Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, *Fundamentals of Data Structures in C*, Silicon Press, ISBN: 0929306406.
4. Richard F. Gilberg and Behrouz A. Forouzan, *Data Structures: A Pseudocode Approach With C*, Thomson Brooks/Cole Publications, Course Technology, ISBN: 9780534390808.
5. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, *Data Structure using C*, Prentice- Hall, ISBN: 9780131997462.
6. Robert Kruse, Tondo C L and Bruce Leung, *Data Structures & Program Design in C*, Pearson India, 2nd Edition, ISBN: 9788177584233.
7. U. A. Deshpande and O. G. Kakde, *Data Structures & Algorithms*, ISTE Learning Materials Centre, New Delhi, ISBN: 9788188057054.
8. Thomas H Cormen, Charles E Leiserson, and Ronald L Rivest, *Introduction to Algorithms*, 3rd Edition, Prentice Hall of India Private Limited, New Delhi, ISBN: 978-0262033848.
9. Seymour Lipschutz, *Data Structures With C*, 1st Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 0070701989.
10. Jean-Paul Tremblay, Paul G. Sorenson, P. G. Sorenson, *Introduction to Data Structures with Applications*, 2nd Edition, Mcgraw-Hill College, ISBN: 0070651574.

Curriculum Plan

Unit/hours (time required)	Topics to be taught (input)	Procedure (process) Student centric Method of teaching	Activity	Learning outcome (output)	Assessment
Unit I	Data structure - definition - types & operations, characteristics - Abstract Data Type – algorithms concepts – quality and complexity of an algorithm -	• Lecture using ICT tools • Problem Solving	Peer group discussion on different algorithm	To Understand • Concept of data structure • Criteria for choosing good algorithms	• Assignment • MCQ evaluation
Unit II	Counting Techniques: asymptotic behaviour of functions. Linear data structures - Arrays records - sorting algorithms - linear search & binary search Linked lists - Stack -s. Recursion - Queue - recursive lists, heterogeneous lists, deterministic skip lists, doubly linked lists and circular lists sparse matrix-representation.	• Lecture using ICT Tols • Illustrations • Hands on sessions	• Seminar • Group study	To Understand • Counting techniques • Linear data structures • Recursion • Linked list	• Lab tests • Q&A exams
Unit III	Non-linear Data Structures - trees - terminology - tree traversals	• Lecture using ICT tools • Illustrati	• Hands on sessions • Illustrations	To Understand • Non linear	• Lab tests • Q&A exams

	algorithms Binary trees - threaded binary trees - binary search trees - traversals and operations on BST heap Tree - balanced trees - M-way trees - B and B+ trees, Red Black Tree, Digital Search Tree, Tries, Treaps, Huffman algorithm for extended binary tree - operations and their implementation. Graphs - representation of graphs – operations - traversals and their implementation	ons • Lab sessions		data structures • Huffman Algorithm	
Unit IV	Hashing - - linear probing - quadratic probing - double hashing - rehashing - extendable hashing - separate chaining - hashing efficiency - heaps -	• Lecture using ICT tools • Illustrations • Hands on Session	• Hands on sessions • Illustrations	To understand • Hashing and its types	• Lab tests • Q&A exams
Unit V	Heap structures - Deaps - leftist heaps - binomial heaps - Fibonacci heaps -binary heaps - skew heaps - pairing heaps - Binomial queues - skew	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand • Heap data structure • Binomial queues • Splay	• Assignments

	heaps - Fibonacci heaps - Splay trees.			trees	

Course Outcomes:

CO:1	Summarize different categories of data structures.
CO:2	CO2: Design algorithms to perform operations with linear and non – linear data structures.
CO:3	CO3: Describe how arrays, linked lists, stacks, queues, trees and graphs are represented in memory and used by algorithms.
CO:4	CO4: Describe common applications for arrays, linked lists, stack, queue, tree and graphs.
CO:5	CO5: Demonstrate different methods for traversing trees.
CO:6	CO6: Design and implement an appropriate hashing function for an application
CO:7	CO7: Discuss the computational efficiency of the principal algorithms for sorting, searching and hashing.
CO:8	CO8: Describes various types of trees and heap structures

Faculty in Charge:

Sindhu T

Programme: CSS1C03 – THEORY OF COMPUTATION

No. of lecture hours per week	4 Hrs
No.of credits for Theory	4

Aim of the course

To provide the students with an understanding of basic concepts in the theory of computation.

Course Outline

Unit I: Preliminaries - Introduction to formal proof and inductive proofs - The central concepts of Automata Theory - Alphabets, Strings, .Languages - Introduction to automata and grammar - Deterministic Finite Automata, Non-deterministic Finite Automata - Equivalence of Deterministic and Nondeterministic Finite Automata - Finite Automata with Epsilon Transitions - Equivalence of NFA with and without epsilon moves.

Unit II: Regular Expressions, Finite Automata and Regular Expressions, Properties of Regular Languages - Pumping lemma and proof for existence of non-regular languages, Closure properties, homomorphism, substitution - Decision Properties - Equivalence and Myhill Nerode and DFA state minimization - Regular Grammar.

.

Unit III: Context Free Languages - Equivalence of CFG and PDA - Normal forms (CNF and GNF) - Closure properties of CFL's - DCFL's and their properties - Decision procedures - CYK algorithm - Pumping lemma and proof for existence of non-context - free languages - Context sensitive languages: Equivalence of LBA and Context Sensitive Grammar (CSG).

.

Unit IV: Turing machines - TM computations - Equivalence of standard TM with multitape and non deterministic TM's - Turing acceptable, Turing decidable and Turing enumerable language classes - Equivalence of type 0 grammars with TM's - Church thesis -Chomsky hierarchy - Closure properties of recursive and recursively enumerable languages.

.

Unit V: Computability and Decidability - halting problem - reductions – post correspondence problem. Computational complexity - Time and space bounded simulations
- Classes P and NP - NP completeness - Cook's theorem.

References:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory, Languages of Computation, 3rd Edition, Prentice Hall, ISBN: 0321455363.
2. Linz P, An Introduction to Formal Languages and Automata, Narosa Publishing House Pvt. Ltd., New Delhi, ISBN: 9788173197819.
3. Michael Sipser, Introduction to Theory of Computation, Cengage Learning India Private Limited, Indian Edition, ISBN: 8131505138.
4. H.R. Lewis and C.H. Papadimitriou, Elements of Theory of Computation, 2nd Edition, Prentice Hall, ISBN: 0132624788.
5. J. E. Savage, Models of Computation, Exploring the Power of Computing, Addison Wesley, 1998, Available at http://cs.brown.edu/~jes/book/.
6. Martin J.C, Introduction to Languages and Theory of Computation, Tata McGraw Hill, 3rd Edition, ISBN: 9780070660489.

Curriculum Plan

Unit/Hours (time required)	Topics to be Taught (Input)	Procedure (process) student centric method of	Activity	Learning outcome (outcome)	Assessment

		teaching			
Unit1(18 Hour)	Preliminaries - Introduction to formal proof and inductive proofs - The central concepts of Automata Theory - Alphabets, Strings, .Languages - Introduction to automata and grammar - Deterministic Finite Automata, Non-deterministic Finite Automata - Equivalence of Deterministic and Nondeterministic Finite Automata - Finite Automata with Epsilon Transitions - Equivalence of NFA with and without epsilon moves.	• Lecture using ICT tools	Peer group discussion on different types of sets	To Understand • Concept of DFA and NFA	• Assignment • MCQ evaluation
Unit 2 (20 Hours)	Regular Expressions, Finite Automata and Regular Expressions, Properties of Regular Languages - Pumping lemma and proof for existence of non-regular languages, Closure properties, homomorphism, substitution - Decision Properties - Equivalence and Myhill Nerode and DFA state minimization - Regular Grammar.	• Lecture using ICT Tools • Illustrations • Hands on sessions	• Seminar • Group stud	To Understand • Regular languages • Pumping Lemma • Regular Grammer .	• Q&A exams.

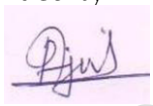
Unit3 (18 Hours)	Context Free Languages - Equivalence of CFG and PDA - Normal forms (CNF and GNF) - Closure properties of CFL's - DCFL's and their properties - Decision procedures - CYK algorithm - Pumping lemma and proof for existence of non-context - free languages - Context sensitive languages: Equivalence of LBA and Context Sensitive Grammar (CSG).	• Lecture using ICT tools • Illustrations	• Hands on sessions • Illustrations	To Understand CFL and CFG	• Q&A exams
Unit4 (14 Hours)	Turing machines - TM computations - Equivalence of standard TM with multi tape and non deterministic TM's - Turing acceptable, Turing decidable and Turing enumerable language classes - Equivalence of type 0 grammars with TM's - Church thesis - Chomsky hierarchy - Closure properties of recursive and recursively enumerable languages.	• Lecture using ICT tools • Illustrations • Hands on Session	• Hands on sessions • Illustrations	To understand Turing machines	• Q&A exams

Unit5(20 Hours)	Computability and Decidability - halting problem - reductions - post correspondence problem. Computational complexity - Time and space bounded simulations - Classes P and NP - NP completeness - Cook's theorem.	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand Computability and decidability	• Assignments • Exams

Course Outcomes:

CO:1	CO1:Concepts of Automata Theory
CO:2	CO2: Describe Finite Automata
CO:3	CO3: Discuss Regular Expressions.
CO:4	CO4: Describe Context free languages
CO:5	CO5: Discuss Push down Automata
CO:6	CO6: Describe Turing Machine
CO:7	CO7: Discuss Computability and decidability
CO:8	CO8: Describes NP completeness.

Faculty in Charge:



Julie P.A

Programme: CSS1C04 – THE ART OF PROGRAMMING METHODOLOGY

Number of Lecture hours per week: 2Hrs
Number of practical hours: 4Hrs
Number of credits for theory:4
Number of credits for practical:1

Aim of the course

- To learn the art of designing algorithms and flowcharts.
- To introduce the concept of algorithmic approach for solving real-life problems.
- To develop competencies for the design and coding of computer programs.
- To learn designing programs with advanced features of C.

Course Outline

Unit I: Part A: Problem Solving - Flow Chart for Structured Programming – Program Charts System Charts - Variables, data names, programming statements - Flow Chart Symbols - Terminal Symbols - I/O - Comments - Connectors - Process - Decision – Loops - Flow Charts of Fundamental Algorithms (mentioned in Part B). **Part B:** Algorithm Design - Problem Solving Aspect - Top down Design - Formal Conventions Writing Algorithms - Fundamental Algorithms (Discuss the Design of Algorithms only). **Part C:** Program, Characteristics of a good program - Modular Approach - Programming style - Documentation and Program Maintenance - Compilers and Interpreters - Running and Debugging Programs - Syntax Errors - Run-Time Errors - Logical Errors - Concept of Structured Programming.

Unit II: Introduction to C Programming - overview and importance of C - C Program Structure and Simple programs - Creation and Compilation of C Programs under Linux and Windows Platforms. Elements of C Language and Program constructs - structure of C program - character set, tokens, keywords, identifier - Data types, constants, symbolic constants, variables, declaration, data input and output, assignment statements. Operators in C - arithmetic operators, relational operators, logical operators, assignment operators,increment and decrement operators, conditional operators, special operators, precedence of operators - arithmetic expressions - evaluation of expressions, type conversion in expressions - precedence and associativity - mathematical functions - I/O operations.

Unit III: Decision making - if statement, if else statement, nesting of if else and else if ladder, switch statement, break statement, continue statement, goto statement, return statement. looping - while, do-while, and for loops, nesting of loops, skipping & breaking loops. Arrays - single dimension arrays - accessing array elements - initializing an array, two dimensional & multi-dimensional arrays - memory representation - strings - processing of strings - string manipulation functions.

Unit IV: The Concept of modularization - defining function - types of functions – User defined functions - function prototype and definition - arguments - passing parameters – call by reference - call by value - returning - nesting of functions and recursion - passing arrays & strings to function - returning multiple values - recursion - scope and life time of variables storage class specifiers - automatic, extern, static storage, register storage. Structures & Union definition, giving values to members, structure initialization, comparison of structure variables, arrays of structures, arrays within structures, structures within arrays, structures and functions, Unions, bit-fields.

Unit V: Pointer - pointer operator - pointer expression - declaration of pointer – initializing pointer - de-referencing - pointer to pointer, constant pointer, array of pointers, pointer to

function. Files - file handling - defining & opening a file - closing a file - Input/output operations on files - error handling, random access to files, command line arguments - dynamic memory allocation - linked lists (concepts only) - preprocessor directives: macro substitution directives - simple macros - macros with arguments - nesting of macros, compiler control directives.

References:

1. Martin M. Lipschutz and Seymour Lipschutz, *Schaum's Outline of Theory and Problems of Data Processing*, ISBN: 9780070379831 (Unit I Part A).
2. Anil Bikas Chaudhuri, *The Art Of Programming Through Flowcharts & Algorithms*, Laxmi Publications, New Delhi (Unit I Part A).
3. Jean Paul Trembley and Pual G Sorenson, *An Introduction to Data Structures with Applications*, Tata McGraw Hill (Unit I Part B).
4. R G Dromey, *How to Solve by Computer*, Pearson Education, 5th Edition, ISBN: 0134340019 (Unit I Part B).
5. J.B Dixit, *Computer Fundamentals and Programming in C*, Firewall Media, ISBN: 8170088828. (Unit I Part C).
6. Dennie Van Tassel, Program Style, *Design, Efficiency, Debugging, and Testing*, PHI, ISBN: 0137299478 (Unit I Part C).
7. E Balagruswamy, *Programming in ANSI C*, TMH, 5th Edition, ISBN: 0070681821.
8. Kamthane, *Programming in C*, 2nd Edition, Pearson India, ISBN: 8131760316.
9. Brian W. Kernighan and Dennis M. Ritchie, *C Programming Language*, PHI, ISBN: 0131103628.
10. Kanetkar, *Let Us C*, BPB Publications, 8th Edition, ISBN: 1934015253.

Curriculum Plan

Unit/hours (time required)	**Topics to be taught (input)**	**Procedure (process) Student centric Method of teaching**	**Activity**	**Learning outcome (output)**	**Assessment**
Unit I	**Part A**: Problem Solving - Flow Chart for Structured Programming – **Part B:** Algorithm Design - Problem Solving Aspect - Top down Design - Formal Conventions Writing Algorithms – **Part C:** Program, Modular Approach - Compilers and	• Lecture using ICT tools • Problem Solving	Peer group discussion on different algorithm	To Understand • Concept Flowchar and algorithms • Modular approach • Errors in programing	• Assignment • Q & A Tests

	Interpreters - Syntax Errors - Run-Time Errors - Logical Errors - Concept of Structured Programming.				
Unit II	Introduction to C ProgrammingOperators in C - evaluation of expressions, type conversion in expressions - precedence and associativity - mathematical functions - I/O operations.	• Lecture using ICT Tols • Illustrations • Hands on sessions	• Seminar • Group study	To Understand • Basics of C programming	• Lab tests • Q&A exams
Unit III	Decision making - if statement, if else statement, nesting of if else and else if ladder, switch statement, break statement, continue statement, goto statement, return statement. looping - while, do-while, and for loops, nesting of loops, skipping & breaking loops. Arrays - single dimension arrays - accessing array elements - initializing an array, two dimensional & multi-dimensional arrays - memory representation - strings - processing of strings -	• Lecture using ICT tools • Illustrations • Lab sessions	• Hands on sessions • Illustrations • Hands on sessions	To Understand • Decision making • Looping • Arrays	• Lab tests • Q&A exams

Unit IV	The Concept of modularization - defining function - types of functions – arguments - passing parameters – call by reference - call by value - recursion - storage class specifiers - Structures & Union	• Lecture using ICT tools • Illustrations • Hands on Session	• Hands on sessions • Illustrations	To understand • Modularisation • Argument passing • Storage class	• Lab tests • Q&A exams
Unit V	Pointer - Files - file handling - error handling, command line arguments - dynamic memory allocation - preprocessor directives:	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand • Pointers in C • File handling • Dynamic memory allocation	• Assignments • Lab tests • Q and A tests

Course Outcomes:

CO:1	Improve ability to develop effective algorithms.
CO:2	Understand the fundamental principles of problem-solving using computers.
CO:3	Demonstrate the applications of the programming constructs including decision making, looping, arrays and strings
CO:4	Conceptualize modular programming basics using functions, structures and Unions
CO:5	Understand features like pointers and macros and to become familiar with programming with files
CO:6	Design, develop, implement, test and document well-structured and reliable computer programs

Faculty in Charge:

Sindhu T

Programme: CSS1C05 – COMPUTER ORGANIZATION & ARCHITECTURE

Number of Lecture hours per week: 4Hrs

Number of credits for theory:4

Objectives: To familiarize with the digital fundamentals, computer organization, computer architecture and assembly language programming.

Course Outline

Unit I: Number systems and Conversions, Boolean Algebra - Truth Tables - Logic gates and Map simplification - flip-flops - design of combinational and sequential circuits - examples of digital circuits - adders, multiplexers, decoders, counters, shift registers – register transfer language and micro operations - data representation - data types, sign and magnitude, complements, fixed-point representation, floating-point representation, other binary codes, error detection codes.

Unit II: Basic computer organization - machine instructions - classification, function, addresses, size, addressing modes - instruction cycle - instruction sequencing. Fundamental concepts - registers, register transfers, performing arithmetic or logic operations, memory read and write, execution of a complete instruction, branch instruction, single bus, two bus, three bus organization, a complete processor - Control unit - hardwired control, micro programmed control, micro instructions-types.

Unit III: Arithmetic & Logic Unit - addition of positive numbers - fast adders – signed addition and subtraction - addition/subtraction logic unit - multiplication of positive numbers - array multiplier, sequential multiplier - signed number multiplication - multiplication using Booth's algorithm - fast multiplication - bit pair recording of multiplication, division-restoring and non-restoring algorithms, floating point numbers and operations.

Unit IV: Main Memory - memory hierarchy - main memory - RAM, ROM - memory cells - cell organization - working - performance considerations - cache memory - virtual memory - memory management requirements - secondary storage - memory interleaving. Input / Output Organization - Accessing I/O ,d&Vices - programmed I/O, interrupt I/O - interrupts - interrupt processing - hardware interrupts - programmable interrupt controller - vectored interrupts - interrupt nesting - daisy chaining - direct memory access (DMA) - DMA operations & DMA Controller, Introduction to I/O interfaces, I/O channels, IO Processors.

Unit V: Architecture - General 8-bit microprocessor and its architecture - 8085 - Functional block diagram - architecture functions of different sections - architecture of 8086 CPU. Instruction Sets - Instruction format - addressing modes - instruction set of 8085 CPU - Instruction cycle - timing diagrams - different machine cycles - fetch and execute operations - estimation of execution time - estimation of execution time. Intel 8051 Micro controller - Architecture - basic instructions - basic assembly language programs peripherals: interrupts, timers, parallel port, serial port.

References:

1. V Carl Hamacher, Zvonko Vranesic and Safwat Zaky, *Computer Organization*, Mc-Graw Hill International Edition, 5th Edition, ISBN: 9780071122184.
2. Morris Mano, *Digital Logic and Computer Design*, Prentice Hall of India, ISBN: 0876924178.
3. M Morris Mano, *Computer System Architecture*, Prentice Hall, 3rd Edition. ISBN: 0131755633.
4. William Stallings, *Computer Organization and Architecture*, 9th Edition, Prentice Hall, ISBN: 013293633X.
5. Andrew S Tanenbaum, *Structured Computer Organization*, Prentice Hall, 6th Edition,ISBN: 0132916525.
6. Floyd Thomas L, *Digital Fundamentals*, Pearson Education, 10th Edition, Prentice Hall, ISBN: 0132359235.
7. Albert Paul Malvino, Donald P Leach, *Digital Principles and Applications*, McGraw Hill, 4th Edition, ISBN: 0070398836.
8. Thomas C Bartee, *Digital Computer Fundamentals*, McGraw Hill, 6th Edition, ASIN:B004H0SL5K.
9. Ramesh. S. Gaonkar, *Microprocessor Architecture*, Programming, and Applications with the 8085, 6th Edition, Wiley Eastern Ltd, New Delhi, ISBN: 9788187972884.
10. Mohamed Rafiquzzaman, *Introduction to Microprocessors and Microcomputer Based System Design*, 2nd Edition, CRC Press, ISBN: 9780849344756.
11. Muhammad Ali Mazidi, Janice Mazidi, Rolin Mckinlay, Janice M. Mazidi, Janice Gillispie Mazidi and Rolin D., *The 8051 Microcontroller and Embedded Systems*, Pearson Education Asia, 5th Indian Reprint, ISBN: 013119402X.

Curriculum Plan

Unit/hours (time required)	**Topics to be taught (input)**	**Procedure (process) Student centric Method of teaching**	**Activity**	**Learning outcome (output)**	**Assessment**
Unit I	Number systems Boolean Algebra flip-flops - design of combinational and sequential circuits - adders, multiplexers, decoders, counters, shift registers – register transfer language and micro operations -	• Lecture using ICT tools • Problem Solvin • Demonstrations	• Peer group discussion on NS • Seminar	To Understand • Number systems • Boolean Algibra • Circuits,Registers	• Assignment • MCQ evaluation

	data representation- binary codes, error detection codes.				
Unit II	Basic computer organization - machine instructions - classification, function, addresses, size, addressing modes - instruction cycle - instruction sequencing. Fundamental concepts - registers, performing arithmetic or logic operations, memory read and write, execution of a complete instruction, branch instruction, bus organization, a complete processor - Control unit -, micro instructions-types.	• Lecture using ICT Tols • Illustrations • Problem solving	• Seminar • Group study	To Understand • Working of machine instructions • Arithemetic and logic operations	• Q&A exams • Assignments
Unit III	Arithmetic & Logic Unit - addition of positive numbers - - array multiplier, sequential multiplier - signed number multiplication -	• Lecture using ICT tools • Illustrations • Problem solving	• Illustrations • Drawing sessions	To Understand • ALU-addition multiplication ,division	• Assignments • Q&A exams

	division-				
Unit IV	Main Memory - memory hierarchy - main memory - RAM, ROM - memory cells - cell organization - working - performance considerations - cache memory - virtual memory- memory interleaving. Input / Output Organization - Accessing I/O devices - interrupts - interrupt processing Introduction to I/O interfaces, I/O channels, IO Processors.	• Lecture using ICT tools • Illustrations •	• Illustrations • Drawing sessions	To understand • Memory hierarchy • Cache and virtual memory • How i/o devices accessed	• Assignments • Q&A exams
Unit V	Architecture - General 8-bit microprocessor and its architecture - 8085 - Intel 8051 Micro controller - interrupts, timers, parallel port, serial port	• Lecture using ICT tools • Illustrations •	• Seminar • Group study	To understand • Microprocessor-8085-8086 • Microcontroller-8051	• Assignments

Course Outcomes:

CO:1	Identify, understand and apply different number systems and codes.
CO:2	Understand the digital representation of data in a computer system.
CO:3	Understand the general concepts in digital logic design and their use in sequential and combinational circuit design.
CO:4	Describe fundamental organization of a computer system

CO:5	Explain addressing modes, instruction formats and program control statements.
CO:6	Understand computer arithmetic formulae and solve problems
CO:7	Distinguish the organization of various parts of a system memory hierarchy.
CO:8	Identify and compare different methods for computer I/O

Faculty in Charge:

Sindhu T

Course: **MSc Computer Science (Second Semester)**

Programme: **CSS2C06 – DESIGN AND ANALYSIS OF ALGORITHMS**

No. of lecture hours per week	4 Hrs
No.of credits for Theory	4

Aim of the course

- To introduce the concept of algorithmic approach for solving real-life problems.
- To teach basic principles and techniques of computational complexity.
- To familiarize with parallel algorithms and related techniques.

Course Outline

Unit I: Algorithm Design: Introduction, Steps in developing algorithm, Methods of specifying an algorithm, Decisions prior to designing: based on the capabilities of the device, based on the nature of solutions, based on the most suitable data structures. Model of Computation: RAM model and PRAM model. Important Problem Types (Introductory concepts): Sorting, Searching, String processing, Graph problems, Combinatorial problems, Geometric problems and Numerical problems.

Unit II: Basic Technique for Design of Efficient Algorithm: Brute Force approach (String matching), Divide-and-Conquer approach (Merge sort), Branch-and-Bound technique (Knapsack problem). Greedy approach (Kruskal's algorithm and Prim's Algorithm), Dynamic Programming (Longest Common Subsequence), Backtracking (Sum of subsets problem).

Unit III: Algorithm Analysis: Importance of algorithm analysis, Time and Space Complexity. Growth of Functions: Asymptotic notations, Cost estimation based on key operations- Big Oh, Big Omega, Little Oh, Little Omega and Theta notations, Big Oh Ratio Theorem, Big Theta Ratio Theorem, Big Omega Ratio Theorem. Analyzing Algorithm Control Structures, Solving Recurrences: Iteration Method, Substitution Method, The Recursion Tree Method, Master's Theorem, Problem solving using Master's Theorem Case 1, Case 2 and Case 3. Analysis of Strasser's algorithm for matrix multiplication, Analysis of Merge sort.

Unit IV: Complexity - Complexity Classes: P, NP, NP Hard and NP Complete problems. NP Completeness reductions for Travelling Salesman Problem and Hamiltonian Cycle. P versus NP problem.

Unit V: Analysing Parallel Algorithms: Time Complexity, Cost, Number of Processors, Space Complexity, Speed up, Efficiency, Scalability, Amdahl's Law. Parallel merging and sorting, Euler tour technique, Parallel prefix computation, Determinist c symmetry breaking.

References:

1. Thomas H Cormen, Charles E Leiserson, and Ronald L Rivest, Introduction to Algorithms, 3rd Edition, Prentice Hall of India Private Limited, New Delhi, ISBN: 9780262033848 (Unit I, II, III and IV).
2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, The Design and Analysis of Computer Algorithms, 1st Edition. Addison Wesley, ISBN: 0534915728 (Unit I, II, III and IV).
3. Pallaw, V K, Design and Analysis of Algorithms, Asian Books Private Ltd, 2012, ISBN: 8184121687 (Unit I, II, III and IV).
4. Sanjay Razdan, Fundamentals of Parallel Computing, Narosa Publishing House, 2014, ISBN: 9788184873481 (Unit V).
5. Pandey H M, Design and Analysis of Algorithms, University Science Press, 2013, ISBN: 9788131803349 (Unit I, II, III and IV).
6. Upadhyay N, Design and Analysis of Algorithms, SK Kataria & Sons, 2008 (Unit I, II, III and IV).
7. U. Manber, Introduction to Algorithms: A Creative Approach, Addison Wesley, ISBN: 9780201003277 (Unit I, II, III and IV).
8. Gilles Brassard and Paul Bratley, Fundamentals of Algorithmics, Prentice-Hall of India, ISBN: 0133350681 (Unit I, II, III and IV).
9. Goodman S E and Hedetniemi, Introduction to the Design and Analysis of Algorithms, Mcgraw Hill, ISBN: 0070237530 (Unit I, II, III and IV).
10. Horowitz E and Sahni S, Fundamentals of Computer Algorithms, Galgotia Publications Pvt. Ltd, ISBN: 8175152575 (Unit I, II, III and IV).
11. Oded Goldreich, P, NP and NP - Completeness, Cambridge University Press, 2011. ISBN: 0521122546 (Unit V).
12. Donald Knuth, The Art of Computer Programming, Fundamental Algorithms, Volume 1, Addison Wesley, 1997, ISBN: 8177587544 (Unit I).
13. Sanjeev Arora and Boaz Borak, Computational Complexity - A Modern Approach, Cambridge University Press; 2009, ISBN: 0521424267 (Unit III).
14. Daniel Hills W and Bruce M Boghosian, Parallel Scientific Computation, Science, 13 August 1993, Vol. 261 (5123), pp.856-863 (Unit V).

Curriculum Plan

Unit/Hours (time required)	Topics to be Taught (Input)	Procedure (process) student centric	Activity	Learning outcome (outcome)	Assessment

		method of teaching			
Unit1(18 Hour)	Algorithm Design: Introduction, Steps in developing algorithm, Methods of specifying an algorithm, Decisions prior to designing: based on the capabilities of the device, based on the nature of solutions, based on the most suitable data structures. Model of Computation: RAM model and PRAM model. Important Problem Types (Introductory concepts): Sorting, Searching, String processing, Graph problems, Combinatorial problems, Geometric problems and Numerical problems. .	• Lecture using ICT tools	Peer group discussion on different types of sets	To Understand • Concept of Algorithm designs	• Assignment • MCQ evaluation
Unit 2 (20 Hours)	Basic Technique for Design of Efficient	• Lecture using ICT Tools • Illustrations	• Seminar • Grou	To Understand • Basic	• Q&A exams.

	Algorithm: Brute Force approach (String matching), Divide-and-Conquer approach (Merge sort), Branch-and-Bound technique (Knapsack problem). Greedy approach (Kruskal's algorithm and Prim's Algorithm), Dynamic Programming (Longest Common Subsequence), Backtracking (Sum of subsets problem).	• Hands on sessions	p stud	techniq ue for design of efficien t algorith m .	
Unit3 (18 Hours)	Algorithm Analysis: Importance of algorithm analysis, Time and Space Complexity. Growth of Functions: Asymptotic notations, Cost estimation based on key operations- Big Oh, Big Omega, Little Oh, Little Omega and Theta notations, Big Oh Ratio Theorem, Big	• Lecture using ICT tools • Illustrations	• Hand s on sessio ns • Illustr ations	To Understand Time and Space complexity	• Q&A exams

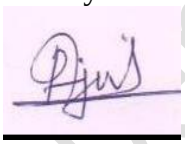
	Theta Ratio Theorem, Big Omega Ratio Theorem. Analyzing Algorithm Control Structures, Solving Recurrences: Iteration Method, Substitution Method, The Recursion Tree Method, Master's Theorem, Problem solving using Master's Theorem Case 1, Case 2 and Case 3. Analysis of Strasser's algorithm for matrix multiplication, Analysis of Merge sort.				
Unit4 (14 Hours)	Complexity - Complexity Classes: P, NP, NP Hard and NP Complete problems. NP Completeness reductions for Travelling Salesman Problem and Hamiltonian Cycle. P versus NP problem.	• Lecture using ICT tools • Illustrations • Hands on Session	• Hands on sessions • Illustrations	To understand P,NP ,NP Hard	• Q&A exams

Unit5(20 Hours)	Analysing Parallel Algorithms: Time Complexity, Cost, Number of Processors, Space Complexity, Speed up, Efficiency, Scalability, Amdahl's Law. Parallel merging and sorting, Euler tour technique, Parallel prefix computation, Deterministi c symmetry breaking.	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand Parallel prefix computation	• Assignments • Exams

Course Outcomes:

CO:1	CO1:Concepts of Algorithm Design
CO:2	CO2: Describe Basic Technique for Design of Efficient Algorithm
CO:3	CO3: Discuss Algorithm Analysis
CO:4	CO4: Describe Recursion Tree method and Masters method
CO:5	CO5: Learn Strassen's matrix multiplication
CO:6	CO6: Describe Complexity
CO:7	CO7: Knowledge about Parallel Algorithms
CO:8	CO8: Understand Parallel prefix computation.

Faculty in Charge:



Julie P.A

Programme:CSS2C07 – OPERATING SYSTEM CONCEPTS COURSE

Number of Lecture hours per week: 3Hrs	Number of practical hours: 3Hrs
Number of credits for theory:4	Number of credits for practical:1

Aim of the Course

- Introduce the underlying principles of an operating system.
- Exposure of multi programming, virtual memory and resource management concepts.
- Case study of public and commercially available operating systems

Course Outline

Unit I: Operating System Overview - Objectives and functions - Evolution of Operating System- Major Achievements - Process Description and Control - Process, Creation & Termination of Processes, Five State Model, Suspended Process, Process Description,Process Control - Modes of Execution, Process Creation, Process and Mode Switching. Threads - Processes Vs Threads, Multithreading, Thread States, Types of Threads, MultiCore and Multithreading. Case Study - Unix SVR4 Process Management, Linux Process and Thread Management.

Unit II: Concurrency - Principles, Race Condition, Operating System Concerns, Process Interaction, Completion for Resources, Cooperation by Sharing. Mutual Exclusion - Requirements, Hardware Support, Semaphores, Producer Consumer Problem, Monitors,Message Passing, Readers/Writers Problem. Deadlock - Principles, Prevention, Avoidance, Detection, Recovery, Dining Philosophers Problem. Case Study: Unix Concurrency Mechanisms.

Unit III: Memory Management, Address binding, Logical Vs Physical address space, Dynamic Loading, Dynamic Linking and Shared Libraries, Overlays, Swapping, Contiguous Memory allocation, Paging, Segmentation, Virtual memory, Demand paging, Page replacement, Thrashing. Case Study: Windows Memory Management.

Unit IV: Uniprocessor Scheduling - types, scheduling algorithms - criteria, nonpreemptive,preemptive. Comparative study of scheduling algorithms - FCFS, SJF, Priority, RR, Multilevel, Feedback Queue. Multiprocessor Scheduling - Classification, Granularity,Design Issues, Process Scheduling, Thread Scheduling. Real Time Scheduling - Background, Characteristics of Real Time OS, Scheduling, Deadline Scheduling, Rate Monotonic Scheduling, Priority Inversion. Case study: Linux Scheduling .

Unit V: Client/Server Computing - Definition, Applications, Classes, Three-Tier Client/Server Architecture, Middleware. Service-Oriented Architecture- Distributed Message Passing - Remote Procedure Calls - Clusters. Mobile Operating Systems - Characteristics - Comparative Study of the Features of iOS and Android.

Reference

1. William Stallings, *Operating System- Internals and Design Principles*, 7th Edition, Pearson, ISBN: 9780273751502.

2. Abraham Silberschatz, Peter B. Galvin and, Greg Gagne, *Operating System Concepts*, 9th Edition, John Wiley & SonsTISBN: 9781118063330.
3. Ann Mclver McHoes and Ida M. Flynn, *Understanding Operating Systems*, 6th Edition, Cengage Learning, 2010, ISBN: 9781439079201.
4. Mukesh Singhal and Niranjan G. Shivaratri, *Advanced Concepts in Operating Systems - Distributed, Database, and Multiprocessor Operating Systems*, Tata McGraw-Hill Education Private Limited, ISBN: 9780070575721.
5. Current Literature (for Mobile Operating Systems).

Curriculum Plan

Unit/hours (time required)	**Topics to be taught (input)**	**Procedure (process) Student centric Method of teaching**	**Activity**	**Learning outcome (output)**	**Assessment**
Unit I(18 Hrs)	OS Overview - Evolution of OS Process, Five State Model, Case Study - Unix SVR4 Process Management, Linux Process and Thread Management.	• Lecture using ICT tools • Problem Solving	• Peer group discussion • Seminar	To Understand • Process and thread • Process models • Process management	• Assignment • MCQ evaluation
Unit II (18 Hrs)	Concurrency - Mutual Exclusion - Semaphores, Producer Consumer Problem, Readers/Writers Problem. Deadlock - Dining Philosophers Problem. .	• Lecture using ICT Tols • Illustrations • Lab sessions	• Seminar • Group study • Hands on sessions	To Understand • Concurrency • ME and semaphores • IPC problems and dead lock	• Lab tests • Q&A exams
Unit III	Memory Management, Address binding,	• Lecture using	• Hands on	To Understand	• Lab tests • Q&A exams

	Logical Vs Physical address space, Dynamic Loading, Contiguous Memory allocation, Paging, Segmentation, Virtual memory, Demand paging, Page replacement, Thrashing.	ICT tools • Illustrations • Lab sessions	sessions • Illustrations	• Memory management • Memory allocation	
Unit IV(18 Hrs)	Uniprocessor Scheduling - Multiprocessor Scheduling - Granularity, Thread Scheduling. Real Time OS, Scheduling, Deadline Scheduling, Rate Monotonic Scheduling,	• Lecture using ICT tools • Illustrations • Hands on Session	• Hands on sessions • Illustrations	To understand • Scheduling • Thread scheduling	• Lab tests • Q&A exams
Unit V(18 Hrs)	Client/Server Computing - Definition, Applications, Classes, Three-Tier Client/Server Architecture, Middleware. Service-Oriented Architecture-Distributed Message Passing - Remote Procedure Calls - Clusters. Mobile Operating Systems -	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand • Cs computing • Middle ware • Android and iOS	• Assignments

	Characteristics - Comparative Study of the Features of iOS and Android.				

Course Outcomes:

CO:1	Explain the basics of database management system, concepts of relational data model, entity- relationship model, relational database design, relational algebra and calculus
CO:2	Apply the normalization techniques to improve the database design
CO:3	Describe various database manipulation commands in SQL.
CO:4	Understand Transaction Processing & Locking using the concept of Concurrency control.
CO:5	Conceptualize advanced features of Object-Oriented Database Management Systems and Distributed databases

Faculty in Charge:

Sindhu T

Programme: **CSS2C08 – COMPUTER NETWORKS**

No. of lecture hours per week	4 Hrs
No.of credits for Theory	4

Aim of the course

- To provide the student with a top down approach of networking starting from the application layer.
- To introduce computer networking in the back drop of Internet protocol stack.

Course Outline

Unit I:Introduction to Computer networks - introduction - topology - categories of networks Internetwork - Internet - network modes- layered model - OSI and TCP/IP Models Transmission media - Wired and unwired media. Computer networks and Internet – the network edge - the network core - network access - delay and loss - protocol layers and services - history of computer networking and Internet.

Unit II: Application layer protocols – principles – the web and HTTP – FTP – Email in Internet – DNS. Socket programming – building a Web server - content distribution.

Unit III: Transport layer services – introduction – relationship between Transport and Network layer – UDP – reliable data transfer – TCP - congestion control - Network layer services – routing – IP - routing in Internet - router - IPV6 - multicast routing – mobility.
.
Unit IV: Link layer services - error detection and correction - multiple access protocols – LAN address – ARP – Ethernet – hubs – bridges – switches - wireless links – PPP - ATM.
.
Unit V: Security in Networks – Principles of Cryptography – Authentication – Integrity –Key Distribution and Certification – Firewalls – Attacks and Counter Measures.

References:

1. J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach Featuring Internet, 6th Edition, Perason Education, ISBN: 0132856204.
2. Behrouz Forouzan, Data Communications and Networking, 4th Edition, McGraw-Hill Reprint, ISBN: 0073250325.
3. Peterson L.L. and Davie B .S., Computer Networks, A Systems Approach, 5th Edition, Morgan Kaufmann, ISBN: 9780123850591.
4. Keshav, An Engineering Approach to Computer Networking, Pearson Education Asia, ISBN: 97898123598652000.
5. Andrew S. Tanenbaum, Computer Networks, 5th Edition, PHI, ISBN: 9788131787571.
6. Herbert Scheldt, Java Complete Reference, 7th Edition, McGraw-Hill Osborne Media, ISBN: 9780072263855.

6. Martin J.C, Introduction to Languages and Theory of Computation, Tata McGraw Hill, 3rd Edition, ISBN: 9780070660489.

Curriculum Plan

Unit/Hours (time required)	**Topics to be Taught (Input)**	**Procedure (process) student centric method of teaching**	**Activity**	**Learning outcome (outcome)**	**Assessment**

Unit1(18 Hour)	Introduction to Computer networks - introduction - topology - categories of networks Internetwork - Internet - network modes- layered model - OSI and TCP/IP Models Transmission media - Wired and unwired media. Computer networks and Internet - the network edge - the network core - network access - delay and loss - protocol layers and services - history of computer networking and Internet.	• Lecture using ICT tools	Peer group discussion on different types of sets	To Understand • Concept of OSI and TCP/IPs	• Assignment • MCQ evaluation
Unit 2 (20 Hours)	Application layer protocols – principles – the web and HTTP – FTP – Email in Internet – DNS. Socket programming – building a Web server - content distribution.	• Lecture using ICT Tools • Illustrations • Hands on sessions	• Seminar • Group stud	To Understand • Application layer protocols .	• Q&A exams.

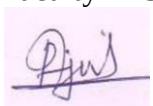
Unit3 (18 Hours)	Transport layer services – introduction – relationship between Transport and Network layer – UDP – reliable data transfer – TCP - congestion control - Network layer services – routing – IP - routing in Internet - router - IPV6 - multicast routing – mobility.	• Lecture using ICT tools • Illustrations	• Hands on sessions • Illustrations	To Understand Transport layer protocols	• Q&A exams
Unit4 (14 Hours)	Link layer services - error detection and correction - multiple access protocols – LAN address – ARP – Ethernet – hubs – bridges – switches - wireless links – PPP - ATM.	• Lecture using ICT tools • Illustrations • Hands on Session	• Hands on sessions • Illustrations	To understand Link Layer services	• Q&A exams

Unit5(20 Hours)	Security in Networks – Principles of Cryptography – Authentication – Integrity – Key Distribution and Certification – Firewalls – Attacks and Counter Measures.	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand Principles of security	• Assignments • Exams

Course Outcomes:

CO:1	CO1:Concepts of Computer networks
CO:2	CO2: Learn about Network Topologies
CO:3	CO3: Discuss Application layer protocols
CO:4	CO4: Describe Transport layer Protocols
CO:5	CO5: Understand Link layer services
CO:6	CO6: Describe Ethernet ,Hub,switches
CO:7	CO7: Discuss Security in Networks
CO:8	CO8: Understand Principles of Cryptography

Faculty in Charge:




Julie P.A , Sindhu T

Programme: CSS2C09 – COMPUTATIONAL INTELLIGENCE

Number of Lecture hours per week: 4Hrs
Number of credits for theory:4

Aim of the course

- To introduce concepts of Artificial Intelligence and Machine Learning.

Course Outline

Unit I: Introduction - Artificial Intelligence - problems, scope and applications, problem space and search - production system- characteristics - the predicate calculus, inference rules,structures and strategies for state space search, strategies for space search, using state space to represent reasoning with the predicate calculus.

Unit II: Heuristics Search: control and implementation of state space search, generate and test, hill climbing, Best-first search, problem reduction, constraint satisfaction, means- ends analysis, heuristic in games, complexity issues.

Unit III: Knowledge representation issues, representation and mappings, representing simple facts in logic, representing instances and ISA relationships, computable functions and predicates, resolution, natural deduction, knowledge representation using rules, logic programming, forward versus backward reasoning, symbolic reasoning under uncertaintynon- monotonic reasoning, depth first search, breadth first search.

Unit IV: Game playing - the mini-max search procedure, adding alpha-beta cut-offs,additional refinement, iterative deepening, planning system and its components,understanding, understanding as constrained satisfaction. Slot and filler structures: semantic nets, frames, conceptual dependency, scripts. Definition and characteristics of expert system, representing and using domain knowledge, expert system shells. Knowledge engineering, knowledge acquisition, expert system life cycle & expert system tools, MYCIN & DENDRAL examples of expert system.

Unit V: Machine learning - rote learning, learning by taking advice, learning in problem solving, learning from examples, explanation based learning, analogy, formal learning theory,connectionist models - hopfield networks, learning in neural networks, back propagation, the genetic algorithm, classifier systems and genetic programming, artificial life and society based learning.

References:

1. Elaine Rich, Kevin Knight and Shivshankar B. Nair, *Artificial Intelligence*, 3rd Edition, Tata - McGraw Hill, New Delhi, ISBN: 0070087709.
2. V S Janakiraman, K Sarukesi and P Gopalakrishnan, *Foundations of Artificial Intelligence and Expert System*, Macmillan India Limited, ISBN: 0333926250.
3. Stuart Russell and Peter Norvg, *Artificial Intelligence: A Modern Approach*, 3rd Edition, Prentice Hall, ISBN: 0136042597.'
4. G. F. Luger and W.A Stubblefield, *Artificial Intelligence - Structures and Strategies for Complex Problem Solving*, Addison-Wesley, 6th Edition, ISBN: 9780321545893.
5. P. H. Winston, *Artificial Intelligence*, Addison-Wesley, 3rd Edition, ISBN: 0201533774.
6. Nils J. Nilsson, Artificial Intelligence, A New Synthesis, 1st Edition, Morgan Kaufmann Publishers, Inc, ISBN: 1558604677.

Curriculum Plan

Unit/hours (time required)	Topics to be taught (input)	Procedure (process) Student centric Method of teaching	Activity	Learning outcome (output)	Assessment
Unit I(18 Hrs)	Introduction - Artificial Intelligence - problems, scope and applications, problem space and search - production system- the predicate calculus,	• Lecture using ICT tools • Problem Solving	• Peer group discussion • Problem solving	To Understand • Concept AI • Production system	• Assignment • MCQ evaluation
Unit II(24 Hrs)	Heuristics Search: state space search, generate and test, hill climbing, Best-first search, problem reduction, constraint satisfaction, means- ends analysis, heuristic in games,	• Lecture using ICT Tols • Illustrations	• Seminar • Group study	To Understand • Various heruristic searches • Heuristic in games	• Lab tests • Q&A exams
Unit III(20 Hrs)	Knowledge representation representing instances and ISA relationships, resolution, natural deduction, knowledge representation using rules, logic programming, forward versus backward reasoning, symbolic reasoning under	• Lecture using ICT tools • Illustrations •	• Illustrations • Problem solving	To Understand • Knowledge representation • Logic programming	• Lab tests • Q&A exams

	uncertainty				
Unit IV(18Hrs)	Game playing - the mini-max search procedure, adding alpha-beta cut-offs, planning system understanding, Slot and filler structures: semantic nets, frames, conceptual dependency, scripts. Definition and characteristics of expert system,	• Lecture using ICT tools • Illustrations •	• Illustrations • Seminar	To understand • Game playing • Knowledge representation	• Lab tests • Q&A exams
Unit V(10 Hrs)	Machine learning - learning, connectionist models -, learning in neural networks, back propagation, the genetic algorithm, classifier systems and genetic programming, artificial life and society based learning	• Lecture using ICT tools • Illustrations	• Seminar • Group study • Seminar	To understand • Machine learning • Genetic Algorithm	• Assignments • Q & A tests

Course Outcomes:

CO:1	Apply the basic principles, models, and algorithms of AI to recognize, model, and solve problems in the analysis and design of information systems
CO:2	Conceptualize various knowledge representation techniques.
CO:3	Analyze the problem-solving methods and algorithms related to searching, reasoning, game playing and machine learning
CO:4	Understand the functioning of expert systems and its importance .
CO:5	Demonstrate the implementation various AI algorithms to solve real life problems

Faculty in Charge:

Sindhu T

Programme: **CSS2C10 – PRINCIPLES OF SOFTWARE ENGINEERING**

No. of lecture hours per week	4 Hrs
No.of credits for Theory	4

Aim of the course

To develop familiarity with software engineering principles and practices.
To have an understanding about the process of product/literature survey, techniques of problem definition, and methods of report writing.

Course Outline

Unit I:Introduction – problem domain - software engineering challenges – approaches –software process and development models – agile models – SDLC - software process.

Unit II: Software requirements analysis & specification - feasibility study - types of feasibility – software requirements - problem analysis – requirement specification –functional specification – metrics. Software design – outcome – cohesion and coupling –layered arrangement of modules – approaches to software design - structured analysis – DFD – extending DFD technique for applying to real-time systems – structured design – detailed design - object oriented modelling – use case model – class diagram – interaction diagram -activity diagram - data diagram – state chart diagram - ER diagram.

Unit III: User Interface (UI) design – characteristics – basic concepts – types –fundamentals of component-based GUI Development – UI design methodology – process planning – cost estimation – project scheduling – configuration management – risk management - software coding – review – documentation – software testing - software testing basics - steps involved in test plan - software testing strategies.

Unit IV: Managing project – time management – setting aims and objectives – techniques for generating ideas – literature survey – types of information sources – writing literature survey.

Unit V: Project story preparation – key deliverables – communicating with experts – forms of communication – presenting ideas – common problems faced by a research scholar – report writing.

References:

1. Pankaj Jalote, An Integrated Approach to Software Engineering, 3rd Edition, Narosa Publishing House, ISBN: 9788173197024.
2. Rajib Mall, Fundamentals of Software Engineering, 3rd Edition, PHI Learning Pvt Ltd, ISBN: 9788120338197.
3. Rohit Khurana, Software Engineering: Principles and Practices, 2nd Edition, Vikas Publishing House Pvt Ltd, ISBN: 8125939466.

4. Andy Hunt, Your Research Hunt, How to Manage it, Routledge, ISBN: 0415344085.
5. Michael Jay Polonsky, David S. Waller, Designing and Managing a Research Project: A Business Student's Guide, Sage, ISBN: 1412977754.
6. Richard Bullock, Maureen Daly Goggin and Francine Weinberg, The Norton Field Guide to Writing (with Readings and Handbook), 3rd Edition, W. W. Norton & Company, ISBN: 0393919595.
7. Kavadia Garg, Agrawal and Agrawal, An introduction to Research Methodology, Rbsa Publishers ISBN: 8176111651.

Curriculum Plan

Unit/Hours (time required)	**Topics to be Taught (Input)**	**Procedure (process) student centric method of teaching**	**Activity**	**Learning outcome (outcome)**	**Assessment**
Unit1(18 Hour)	Introduction – problem domain - software engineering challenges – approaches – software process and development models – agile models – SDLC - software process.	• Lecture using ICT tools	Peer group discussion on different types of sets	To Understand • Concept of SDLC	• Assignment • MCQ evaluation
Unit 2 (20 Hours)	Software requirements analysis & specification - feasibility study - types of feasibility – software requirements - problem analysis – requirement specification – functional specification – metrics. Software design – outcome – cohesion and	• Lecture using ICT Tools • Illustrations • Hands on sessions	• Seminar • Group stud	To Understand • SRS • DFD • ER Diagram • Activity Diagram.	• Q&A exams.

	coupling – layered arrangement of modules – approaches to software design - structured analysis – DFD – extending DFD technique for applying to real-time systems – structured design – detailed design - object oriented modelling – use case model – class diagram – interaction diagram - activity diagram - data diagram – state chart diagram - ER diagram.				
Unit3 (18 Hours)	User Interface (UI) design – characteristics – basic concepts – types – fundamentals of component-based GUI Development – UI design methodology – process planning – cost estimation – project scheduling – configuration management – risk	• Lecture using ICT tools • Illustrations	• Hands on sessions • Illustrations	To Understand User Interface	• Q&A exams

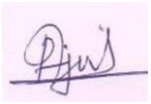
	management - software coding – review – documentation – software testing - software testing basics - steps involved in test plan - software testing strategies.				
Unit4 (14 Hours)	Managing project – time management – setting aims and objectives – techniques for generating ideas – literature survey – types of information sources – writing literature survey.	• Lecture using ICT tools • Illustrations • Hands on Session	• Hands on sessions • Illustrations	To understand Project management	• Q&A exams
Unit5(20 Hours)	Project story preparation – key deliverables – communicating with experts – forms of communication – presenting ideas – common problems faced by a research scholar – report writing.	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand Report writing	• Assignments • Exams

Course Outcomes:

CO:1	CO1:Concepts of Software Engineering
CO:2	CO2: Describe Software processes
CO:3	CO3: Knowledge About process models

CO:4	CO4: Describe object oriented modelling
CO:5	CO5: Understand User interface Design
CO:6	CO6: Describe Software testing
CO:7	CO7: Discuss Project management
CO:8	CO8: Describes Report Writing

Faculty in Charge:



Julie P.A

Course : MSc Computer Science (Third Semester)

Programme: CSS3C11 – ADVANCED DATABASE MANAGEMENT SYSTEM

Number of Lecture hours per week: 3Hrs	Number of practical hours: 3Hrs
Number of credits for theory:4	Number of credits for practical:1

Aim of the Course

Number of Lecture hours per week: 3Hrs

- To understand the relational model, and know how to translate requirements captured in an Entity-Relationship diagram into a relational schema
- To reason about dependencies in a relational schema.
- To understand normal form schemas, and the decomposition process by which normal forms are obtained.
- To familiarize with advanced SQL' statements
- To understand advanced features of database technologies.

Course Outline

Unit I: Introduction - purpose of database systems, views of data - data abstraction, instances and schemas, data independence, data models - hierarchical data model, network data model, relational data model, ER d&tg9,mg9lei. Database languages - DDL, DML, transaction anagement, storage management, database administrator, database users, overall system structure. Relational data model - relational model concepts, keys, integrity constraints - domain constraints, key constraints, entity integrity constraints, referential integrity constraints. ER data model - basic concepts, constraints, keys, design issues, entity relationship diagram, weak entity sets, extended ER features, design of an ER database schema, reduction of an ER schema to tables. Relational algebra and calculus – relational algebra - selection and projection, set

operations, renaming, joins, division. Relational calculus - tuple relational calculus, domain relational calculus. Expressive power of algebra and calculus.

Unit II: Relational database design - anomalies in a database - functional dependency -lossless join and dependency- preserving decomposition - normalization - normal forms -first, second and third normal form - Boyce Codd normal form - multivalued, dependency - fourth normal form - join dependency - project join normal form - domain key normal form.

Unit III: Relational database query languages - basics of QBE and SQL. Data definition in SQL data types, creation, insertion, viewing, updation, deletion of tables, mo difying the structure of the tables, renaming, dropping of tables. Data constraints - I/O constraints, primary key, foreign key, unique key constraints, ALTER TABLE command database manipulation in SQL - computations done on table data - SELECT command, logical operators, range searching, pattern matching, grouping data from tables in SQL, GROUP BY, HAVING clauses. Joins - joining multiple tables, joining a table to it. DELETE - UPDATE. Views - creation, renaming the column of a view, destroys view. Program with SQL - data types Using SET and SELECT commands, procedural flow, IF, IF /ELSE, WHILE, GOTO, global variables. Security - locks, types of locks, levels of locks. Cursors - working with cursors, error handling, developing stored procedures, CREATE, A LTER and
DROP, passing and returning data to stored procedures, using stored procedures within queries, building user defined functions, creating and calling a scalar function, implementing triggers, creating triggers, multiple trigger interaction (Use MySQL as the RDBMS).

Unit IV: Transaction management, concurrency control and query processing- concept,definition and states of transactions, ACID properties - concurrency control, serializability - conflict serializability, view serializability, recoverability-recoverable schedules, noncascading schedules, strict schedules. Concurrency control schemes - locking- two phase locking, deadlock, granularity, timestamp ordering protocol. Basics of query processing.

Unit V: Object Oriented Database Management Systems (OODBMS) - concepts, need for OODBMS, composite objects, issues in OODBMSs, advantages and disadvantages of OODBMS. Distributed databases - motivation - distributed database concepts, types of distribution, architecture of distributed databases, the design of distributed databases, distributed transactions, commit protocols for distributed databases.

Reference

1. Elmasri and Navathe, *Fundamentals of Database Systems*, 5th Edition, Pearson, ISBN: 9788131758984.
2. Abraham Silbersehatz, Henry F. Korth and S.Sudarshan, *Database System Concepts*, 6th Edition, Tata McGraw-Hill, ISBN: 0071325220.
3. CJ Date, *An Introduction to Database Systems*, 8th Edition, Addison Wesley, ISBN: 0321197844.
4. Ramakrishnan and Gehrke, *Database Management Systems*, 3rd Edition, McGraw - Hill Education, ISBN: 9339213114.
5. Alexis Leon and Mathews Leon, *Database Management Systems*, 1st Edition, Vikas Publishers, ISBN: 8182092221.

6. Vikram Vaswani, *MySQL The complete Reference*, 1st Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 0070586845.
7. Joel Murach, *Murach's Mysql*, Mike Murach & Associates Inc, ISBN: 9350237695.
8. Paul DuBois, *MySQL Cookbook*, 2nd Edition, O'Reilly Media, ISBN: 184042809.

Curriculum Plan

Unit/hours (time required)	**Topics to be taught (input)**	**Procedure (process) Student centric Method of teaching**	**Activity**	**Learning outcome (output)**	**Assessment**
Unit I(15 Hrs)	Introduction to DBMS-views of data - data models - Database languages - database users, Relational data model - ER data model - Relational algebra and calculus – relational algebra -	• Lecture using ICT tools • Problem Solving	• Peer group discussion • Seminar	To Understand • DBMS vies • DBMS users • DBMS Languages • DBMS models	• Assignment • MCQ evaluation
Unit II(18 Hrs)	Relational database design - anomalies in a database - functional dependency - lossless join and dependency-preserving decomposition -normalization-	• Lecture using ICT Tols • Illustrations • Lab sessions	• Seminar • Group study • Hands on sessions	To Understand • Anomalies • Dependency preserving • Normalisation	• Lab tests • Q&A exams
Unit III(22 Hrs)	Relational database query languages – SQL query processing	• Lecture using ICT tools • Illustrations • Lab	• Hands on sessions • Illustrations	To Understand • SQL query processing	• Lab tests • Q&A exams

		sessions			
Unit IV(20 Hrs)	Transaction management, - ACID properties -, serializability schedules, Concurrency control schemes - locking- deadlock, granularity, timestamp ordering protocol.	• Lecture using ICT tools • Illustrations • Hands on Session	• Hands on sessions • Illustrations	To understand • Transactin in DBMS • Concurrency and serialisability • Deadlock	• Lab tests • Q&A exams
Unit V(15 Hrs)	Object Oriented Database Management Systems Distributed databases - distributed transactions, commit protocols for distributed databases.	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand • OODBMS • Distributed Databses	• Assignments

Course Outcomes:

CO:1	Explain the basics of database management system, concepts of relational data model, entity- relationship model, relational database design, relational algebra and calculus
CO:2	Apply the normalization techniques to improve the database design
CO:3	Describe various database manipulation commands in SQL.
CO:4	Understand Transaction Processing & Locking using the concept of Concurrency control.
CO:5	Conceptualize advanced features of Object-Oriented Database Management Systems and Distributed databases

Faculty in Charge:

Sindhu T

Programme: **CSS3C12 – Object Oriented Programming Concepts**

No. of lecture hours per week	4 Hrs
No.of credits for Theory	4

Aim of the course

To learn object oriented concepts and programming concepts and methodologies and to learn its implementation using Java.

.

Course Outline

Unit I:Introduction to OOPS - basic principles of object orientation (objects , attributes and methods, encapsulation and information hiding, state retention, object identity, messages, class hierarchy, inheritance, polymorphism, genericity) - introduction to Java - history, versioning, the Java Virtual Machine, byte code, features of Java, language components - primitive data types, comments, keywords, literals, variables scope & declarations, control structures - FOR, IF, WHILE, DO WHILE, SWITCH, BREAK, CONTINUE statements - operators - casts and conversions - arrays.

Unit II: Object - oriented programming – classes - class fundamentals - declaring objects -new operator – methods – parameter passing – constructors - parameterized constructors -this keyword – finalize method. Overloading methods and constructors, access controls, static and final, nested and inner classes. Inheritance - extends, member access and inheritance, super keyword, polymorphism, method overriding, dynamic method dispatch, abstract classes, packages and interfaces.

Unit III: Exceptions, threads & IO in Java - The file and standard streams, stream classes and interfaces, using byte streams and character streams, threads - threads vs. processes, creating threads, runnable interface, thread class, inter thread communication, synchronization. Exceptions - basic of Java exception handling, hierarchy, developing user defined exception classes.

Unit IV: Applets, AWT & Swing - applet class, types of applet, skeleton, applet tag, passing parameters, event handling, delegation event model, event classes, listeners, AWT classes and window fundamentals, frames, working with fonts, graphics and colors, AWT controls, layouts and menus, dialogue boxes. Swings - Japplets, icon, labels, buttons, textbox, combo box, tables and panes.

Unit V: Database and sockets - JDBC - introduction, architecture, drivers, connections, statements, resultset and meta data (Use MySQL as the RDBMS). Sockets: introduction to networking, InetAddress, url, socket, server sockets, datagrams. Introduction to Unified Modelling Language (UML), UML diagrams, class diagrams, object interaction diagrams, state and activity diagrams, component diagrams, deployment diagrams. Introduction to analysis - object oriented system analysis, design and implementations.

.**References:**

1.Herbert Scheldt, Java Complete Reference, 8th Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 1259002462.
2. E Balaguruswamy, Programming in Java: A Primer, 4th Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 007014169X.
3. Kathy Sierra, Head First Java, 2nd Edition, Shroff Publishers and Distributors Pvt Ltd, ISBN: 8173666024.
4. David Flanagan, Jim Farley, William Crawford and Kris Magnusson, Java Enterprise in a Nutshell: A Desktop Quick Reference, 3rd Edition, O'Reilly Media, ISBN: 0596101422.
5. Grady Booch, James Rumbaugh and Ivar Jacobson, The Unified Modeling Language User Guide, 2nd Edition, Pearson, ISBN: 8131715825.

Curriculum Plan

Unit/Hours (time required)	**Topics to be Taught (Input)**	**Procedure (process) student centric method of teaching**	**Activity**	**Learning outcome (outcome)**	**Assessment**
Unit1(18 Hour)	Introduction to OOPS - basic principles of object orientation (objects , attributes and methods, encapsulation and information hiding, state retention, object identity, messages, class hierarchy, inheritance, polymorphism, genericity) - introduction to Java - history, versioning, the Java Virtual Machine, byte	• Lecture using ICT tools	Peer group discussion on different types of sets	To Understand • Concept of OOPS	• Assignment • MCQ evaluation

	code, features of Java, language components - primitive data types, comments, keywords, literals, variables scope & declarations, control structures - FOR, IF, WHILE, DO WHILE, SWITCH, BREAK, CONTINUE statements - operators - casts and conversions - arrays.				
Unit 2 (20 Hours)	Object - oriented programming – classes - class fundamentals - declaring objects- new operator – methods – parameter passing – constructors - parameterized constructors - this keyword – finalize method. Overloading methods and constructors, access controls, static and final, nested and inner classes. Inheritance - extends, member access and inheritance, super	• Lecture using ICT Tools • Illustrations • Hands on sessions	• Seminar • Group stud	To Understand • Objest • Classes • methods • Interfaces.	• Q&A exams.

	keyword, polymorphism, method overriding, dynamic method dispatch, abstract classes, packages and interfaces.				
Unit3 (18 Hours)	Exceptions, threads & IO in Java - The file and standard streams, stream classes and interfaces, using byte streams and character streams, threads - threads vs. processes, creating threads, runnable interface, thread class, inter thread communication, synchronization. Exceptions - basic of Java exception handling, hierarchy, developing user defined exception classes.	• Lecture using ICT tools • Illustrations	• Hands on sessions • Illustrations	To Understand Exception Handling	• Q&A exams
Unit4 (14 Hours)	Applets, AWT & Swing - applet class, types of applet, skeleton, applet tag, passing parameters, event handling,	• Lecture using ICT tools • Illustrations • Hands on	• Hands on sessions • Illustrations	To understand AApplet,AWT,Swing	• Q&A exams

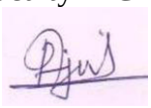
	delegation event model, event classes, listeners, AWT classes and window fundamentals, frames, working with fonts, graphics and colors, AWT controls, layouts and menus, dialogue boxes. Swings - Japplets, icon, labels, buttons, textbox, combo box, tables and panes.	Session			
Unit5(20 Hours)	Database and sockets - JDBC - introduction, architecture, drivers, connections, statements, resultset and meta data (Use MySQL as the RDBMS). Sockets: introduction to networking, InetAddress, url, socket, server sockets, datagrams. Introduction to Unified Modelling Language (UML), UML diagrams, class diagrams, object interaction	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand JDBC	• Assignments • Exams

	diagrams, state and activity diagrams, component diagrams, deployment diagrams. Introduction to analysis - object oriented system analysis, design and implementations.				

Course Outcomes:

CO:1	CO1: Understand the Concepts of OOPS
CO:2	CO2: Describe Features of Java
CO:3	CO3: Discuss Byte code , JVM
CO:4	CO4: Describe object, class
CO:5	CO5: Discuss Exception Handling and Threads
CO:6	CO6: Describe Applet, AWT ,Swing
CO:7	CO7: Discuss Database and Sockets
CO:8	CO8: Knowledge about UML Diagrams

Faculty in Charge:



Julie P.A

Programme: CSS3C13 Principles of Compilers

Number of Lecture hours per week: 3Hrs	Number of practical hours: 3Hrs
Number of credits for theory:4	Number of credits for practical:1

Aim of the Course

To introduce the fundamental concepts and various phases of compiler design.

Course Outline

Unit I: Introduction to compiling - definition of compiler, translator, interpreter, analysis of the source program, the phases of a compiler, compiler construction tools- applications of compiler technology – programming language basics - lexical analysis – role of lexical analyser – input buffering - specification of tokens – recognition of tokens using finite automata - regular expressions and finite automata - from NFA to DFA - Regular Expression to an NFA - Design of a lexical analyser generator.

Unit II: Syntax analysis – role of parser – error handling and recovery – definitions of parsing, top-down parsing and bottom-up parsing - context free grammars – derivations -parse tree – ambiguity – associativity and precedence of operators - writing a grammar – topdown parsing – recursive descent parsing - FIRST and FOLLOW – LL (1) Grammars –recursive predictive parsing - bottom up parsing – reductions – handle pruning – shift reduce parsing - operator precedence parsing, simple LR parsing.

Unit III: Intermediate code generation – DAG – three address code – addresses and instructions – quadruples – triples – Static Simple Assignment form – types and declarations – type expressions - type equivalences – declarations – type checking – rules – type conversion – function and operator overloading – type inference and polymorphic functions – control flow – boolean expressions – short circuit code – flow-control statements – control-flow translation for boolean expressions – BREAK CONTINUE and GOTO statements.

Unit IV: Run time environments – storage optimization – static Vs dynamic allocation –stack allocation of space - activation trees and records – calling sequences – access to non local data on the stack – data access without nested procedures – issues with nested procedures – heap management – the memory manager – the memory hierarchy – locality in programs – reducing fragmentation - manual deallocation requests.

Unit V: Code generation – issues in the design of a code generator – the target language – a simple target machine model – the program and instruction costs – address in the target code – static allocation – stack allocation – run-time address for names – basic blocks and flow graphs – representation of flow graphs. Code optimization - the principal sources of optimization – data flow analysis – abstraction – data flow analysis schema – data flow schemas on basic blocks – reaching definitions – live variable analysis – available expressions. Region based analysis – regions – region hierarchies for reducible flow graphs – overview of a region based analysis.

Reference

1. V Aho A, Ravi Sethi, D Ullman J, *Compilers Principles*, Techniques and Tools, 2nd Edition, Pearson Education Singapore Pte Ltd, ISBN: 8131721019.
2. K. V. N. Sunitha, *Compiler Construction*, Pearson, ISBN:9789332500297.
3. W Appel and Andrew, *Modern Compiler Implementation in C*, 1st Edition,Cambridge University Press, ISBN: 817596071X.
4. Allen I Holub, *Compiler Design in C*, 1st Edition, PHI Learning Pvt Ltd, ISBN: 812030778X.
5. Tremblay and Sorenson, *The Theory and Practice of Compiler Writing*, 1st Edition, BSP Books Pvt Ltd, ISBN: 8178000776.

6. Torben Ægidius Mogensen, *Basics of Compiler Design*, Department of Computer Science, University of Copenhagen (Online Edition).

Curriculum Plan

Unit/hours (time required)	**Topics to be taught (input)**	**Procedure (process) Student centric Method of teaching**	**Activity**	**Learning outcome (output)**	**Assessment**
Unit I(18 Hrs)	Introduction to compiling - definition of compiler, translator, interpreter, the phases of a compiler, finite automata - regular expressions and finite automata - from NFA to DFA - Regular Expression to an NFA - Design of a lexical analyser generator.	• Lecture using ICT tools • Problem Solving	• Peer group discussion • Seminar	To Understand • Compiler and interpreter • Phases of compiler • Lexical Analysis	• Assignment • MCQ evaluation
Unit II(20 Hrs)	Syntax analysis – role of parser – – ambiguity – associativity and precedence of operators topdown parsing – recursive descent parsing - FIRST and FOLLOW – LL (1) bottom up parsing – LR parsing.	• Lecture using ICT Tols • Illustrations	• Seminar • Group study • Problem solving	To Understand • Syntax analysis • Abiguity • Top down-bottom up parser.	• Assignments • Q&A exams
Unit III(20Hrs)	Intermediate code generation – DAG – three address code –	• Lecture using ICT tools • Illustratio	• Illustrations • Prob	To Understand • ICG- • Three	• MCQ tests • Q&A exams

	– type expressions - type equivalences – type boolean expressions – short circuit code – flow-control statements – control-flow translation for boolean expressions – BREAK CONTINUE and GOTO statements.	ns	lem solvi ng	address code • Flow control expressi ons	
Unit IV(17 Hrs)	Run time environments – storage optimization – static Vs dynamic allocation – – heap management – the memory manager – the memory hierarchy – locality in programs – reducing fragmentation - manual deallocation requests.	• Lecture using ICT tools • Illustratio ns	• Illus trati ons • Dra wing	To understand • Storage optimis ation • Memor y hierarch y	• Assignments • Q&A exams
Unit V(15 Hrs)	Code generation Code Region based analysis – regions – region hierarchies for reducible flow graphs –	• Lecture using ICT tools • Illustratio ns • Hands on Session	• Sem inar • Gro up stud y	To understand • CG • Region based analysis	• Assignment s • Q & A tests

Course Outcomes:

CO:1	Understand the major phases of compilation, identify tokens of a typical high -level programming language, define regular expressions for tokens, design and implement a lexical analyzer
CO:2	Develop the parsers and experiment the knowledge of different parsers design without automated tools

CO:3	Construct the intermediate code representations and generation.
CO:4	Explain the role of different types of runtime environments and memory organization for implementation of typical programming languages
CO:5	Apply the optimization techniques to have a better code for code generation

Faculty in Charge:

Sindhu T

Programme: CSS3E01a – COMPUTER GRAPHICS

No. of lecture hours per week	4 Hrs
No.of credits for Theory	4

Aim of the course

- To understand the fundamentals of the modern computer graphics.
- To pipeline the mathematics of affine transformations in three dimensions.
- To understand the common data structures to represent and manipulate geometry,
- colour and light representation and manipulation in graphics systems.
- To have an exposure to programming in Open GL.

Course Outline

Unit I:Introduction – Application of computer graphics, Video Display Devices- refresh CRT, raster and random scan display, color CRT, flat panel, LCD, LED, DVST. Raster -Scan Systems-video controller, display processor, Random-Scan Systems.

Unit II: 2D Graphics: Line drawing algorithms – DDA, Bresenham's – Midpoint Circle drawing algorithm –Filling-Scan line polygon fill algorithm, boundary fill algorithm, floodfill algorithm, 2D Transformations-translation, rotation, scaling, shearing and reflection, composite transformations. 2D Viewing –the viewing pipeline, viewing coordinate reference frame, window-to- viewport coordinate transformation. Clipping-point clipping, Cohen Sutherland line clipping, Sutherland Hodgeman polygon clipping, text clipping.

Unit III: 3D Graphics: 3D Transformations- translation, rotation, scaling, shearing and reflection,3D Viewing-viewing pipeline, viewing coordinates, projections- parallel & perspective projections.

Unit IV: 3D object representation - wireframe model, curve representation, surfaces, spline representation, bezier curves, cubic spline. Visible surface detection methods- classification, back-face detection, Z-buffer algorithm.

Unit V: Discrete Techniques and OpenGL programming - Texture mapping, Bit and Pixel operations, Compositing, Sampling and Aliasing Techniques. Introduction to OpenGL, Features in OpenGL, OpenGL operations, Abstractions in OpenGL - GL, GLU & GLUT, a few examples of OpenGL programs.

References:

1.Donald Hearn and M. Pauline Baker, Computer Graphics, 2nd Edition, Prentice Hall, ISBN: 0135309247.
2. Donald D. Hearn, M. Pauline Baker and Warren Carithers, Computer Graphics with Open GL, 4th Edition, Prentice Hall, ISBN: 9780136053583
3. Hill, Computer Graphics using OpenGL, 3rd Edition, Prentice Hall of India Private Ltd. New Delhi, ISBN: 8120338294.
4. Mason Woo, Jackie Neider, Tom Davis, Dave Shreiner, Dave Shriner and Tom David, Open GL Programming Guide, 6th Edition, Person, ISBN: 9780201604580.
5. The Official Guide to Learning OpenGL, Version 1.1, Available at http://www.glprogramming.com/red/.
6. Shreiner and Angel, Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL, 6th Edition, Pearson Education, ISBN: 0132545233.

Curriculum Plan

Unit/Hours (time required)	**Topics to be Taught (Input)**	**Procedure (process) student centric method of teaching**	**Activity**	**Learning outcome (outcome)**	**Assessment**
Unit1(18 Hour)	Introduction – Application of computer graphics, Video Display Devices- refresh CRT, raster and random scan display, color CRT, flat panel, LCD, LED, DVST. Raster - Scan Systems-	• Lecture using ICT tools	Peer group discussion on different types of sets	To Understand • Application of computer Graphics	• Assignment • MCQ evaluation

	video controller, display processor, Random-Scan Systems.				
Unit 2 (20 Hours)	2D Graphics: Line drawing algorithms – DDA, Bresenham's – Midpoint Circle drawing algorithm – Filling-Scan line polygon fill algorithm, boundary fill algorithm, floodfill algorithm, 2D Transformations-translation, rotation, scaling, shearing and reflection, composite transformations. 2D Viewing –the viewing pipeline, viewing coordinate reference frame, window-to-viewport coordinate transformation. Clipping-point clipping, Cohen Sutherland line clipping, Sutherland Hodgeman polygon clipping, text clipping.	• Lecture using ICT Tools • Illustrations • Hands on sessions	• Seminar • Group stud	To Understand • line drawing algorithm • circle drawing • Transformation • Clipping.	• Q&A exams.

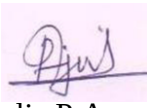
Unit3 (18 Hours)	3D Graphics: 3D Transformations-translation, rotation, scaling, shearing and reflection,3D Viewing-viewing pipeline, viewing coordinates, projections-parallel & perspective projections.	• Lecture using ICT tools • Illustrations	• Hands on sessions • Illustrations	To Understand Transformations	• Q&A exams
Unit4 (14 Hours)	3D object representation - wireframe model, curve representation, surfaces, spline representation, bezier curves, cubic spline. Visible surface detection methods-classification, back-face detection, Z-buffer algorithm.	• Lecture using ICT tools • Illustrations • Hands on Session	• Hands on sessions • Illustrations	To understand 3D object representation	• Q&A exams
Unit5(20 Hours)	Discrete Techniques and OpenGL programming - Texture mapping, Bit and Pixel operations, Compositing, Sampling and Aliasing Techniques. Introduction to OpenGL, Features in	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand OpenGL	• Assignments • Exams

	OpenGL, OpenGL operations, Abstractions in OpenGL - GL, GLU & GLUT, a few examples of OpenGL programs.				

Course Outcomes:

CO:1	CO1: Understand the Applications of computer Graphics
CO:2	CO2: Differentiate LED , LCD ,DVST
CO:3	CO3: Understand 2D Graphics
CO:4	CO4: Gain knowledge of line and circle drawing algorithm
CO:5	CO5: Understand different methods of clipping
CO:6	CO6: Understand Texture mapping
CO:7	CO7: Familiar about Texture mapping
CO:8	CO8: Understand OpenGL Programming

Faculty in Charge:



Julie P.A

Programme: CSS3E02f – DATA WAREHOUSING AND DATA MINING

No. of lecture hours per week	4 Hrs
No.of credits for Theory	4

Aim of the course

To provide the fundamentals on information retrieval and data mining techniques To focus on practical algorithms of textual document indexing, relevance ranking, web usage mining, text analytics, as well as their performance evaluations. To give an exposure to the fundamentals of Data Analytics.

Course Outline

Unit I: Data warehouse - definition - operational database systems Vs data warehouses - multidimensional model - from- tables and spreadsheets to Data Cubes - schemas for multidimensional databases - measures - concept hierarchies - OLAP operations in the multidimensional data model - data warehouse architecture.

Unit II: Data mining - introduction - definition - data mining functionalities - major issues in data mining - data pre-processing - data cleaning - data integration and transformation -data reduction - data discretization and concept hierarchy generation. Association rule mining - efficient and scalable frequent item set mining methods - mining various kinds of association rules - association mining to correlation analysis - constraint- based association mining.

Unit III: Classification and prediction - issues regarding classification and prediction - classification by decision tree introduction - Bayesian classification - rule based classification - classification by back propagation - support vector machines - associative classification - lazy learners - other classification methods - prediction - accuracy and error measures - evaluating the accuracy of a classifier or predictor - ensemble methods - model section.

Unit IV: Cluster analysis - types of data in cluster analysis - a categorization of major clustering methods - partitioning methods - hierarchical methods - density-based methods -grid-based methods - model-based clustering methods - clustering high dimensional data -constraint-based cluster analysis - outlier analysis.

Unit V: Graph mining - mining object, spatial, multimedia, text and web data -multidimensional analysis and descriptive mining of complex data objects - spatial data mining - multimedia data mining - text mining - mining the World Wide Web.

References:

1Jain Pei, Jiawei Han and Micheline Kamber, Data Mining Concepts and Techniques, 3rd Edition, Elsevier, ISBN: 9380931913.
2. Alex Berson and Stephen J. Smith, Data Warehousing, Data Mining & OLAP, Computing Mcgraw-Hill, ISBN: 0070062722.
3. K.P. Soman, Shyam Diwakar and V. Ajay, Insight into Data mining Theory and Practice, 1st Edition, Prentice Hall of India, ISBNy.8120328973.
4. G. K. Gupta, Introduction to Data Mining with Case Studies, 3rd Edition, PHI Learning Pvt. Ltd, ISBN: 8120350022.
5. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Introduction to Data Mining, 1st Edition, Pearson India, ISBN: 9332518653.

Curriculum Plan

Unit/Hours (time required)	Topics to be Taught (Input)	Procedure (process) student	Activity	Learning outcome (outcome)	Assessment

		centric method of teaching			
Unit1(18 Hour)	Data warehouse - definition - operational database systems Vs data warehouses - multidimensional model - from-tables and spreadsheets to Data Cubes - schemas for multidimensional databases - measures - concept hierarchies - OLAP operations in the multidimensional data model - data warehouse architecture.	• Lecture using ICT tools	Peer group discussion on different types of sets	To Understand • Concept of Data ware house	• Assignment • MCQ evaluation
Unit 2 (20 Hours)	Data mining - introduction - definition - data mining functionalities - major issues in data mining - data pre-processing - data cleaning - data integration and transformation - data reduction - data discretization and concept hierarchy generation. Association rule	• Lecture using ICT Tools • Illustrations • Hands on sessions	• Seminar • Group stud	To Understand • Datamining functionalities • Data integration and transformation .	• Q&A exams.

	mining - efficient and scalable frequent item set mining methods - mining various kinds of association rules - association mining to correlation analysis - constraint- based association mining.				
Unit3 (18 Hours)	Classification and prediction - issues regarding classification and prediction - classification by decision tree introduction - Bayesian classification - rule based classification - classification by back propagation - support vector machines - associative classification - lazy learners - other classification methods - prediction - accuracy and error measures - evaluating the accuracy of a classifier or predictor - ensemble methods - model section.	• Lecture using ICT tools • Illustrations	• Hands on sessions • Illustrations	To Understand Classification and prediction	• Q&A exams

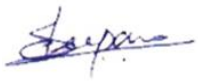
Unit4 (14 Hours)	Cluster analysis - types of data in cluster analysis - a categorization of major clustering methods - partitioning methods - hierarchical methods - density-based methods - grid-based methods - model-based clustering methods - clustering high dimensional data - constraint-based cluster analysis - outlier analysis.	• Lecture using ICT tools • Illustrations • Hands on Session	• Hands on sessions • Illustrations	To understand Cluster analysis	• Q&A exams
Unit5(20 Hours)	Graph mining - mining object, spatial, multimedia, text and web data - multidimensional analysis and descriptive mining of complex data objects - spatial data mining - multimedia data mining - text mining - mining the World Wide Web.	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand Graph mining	• Assignments • Exams

Course Outcomes:

CO:1	CO1: Understand Data Warehouse
CO:2	CO2: Familiar about OLAP
CO:3	CO3: Understand Data mining functionalities

CO:4	CO4: Gain knowledge of Classification
CO:5	CO5: Understand Prediction
CO:6	CO6: Understand Clustering
CO:7	CO7: Familiar about Graph mining
CO:8	CO8: Understand OpenGL Programming

Faculty in Charge:



Deepa M

Course: **MSc Computer Science (Fourth Semester)**

Programma:CSS4E03c-SYSTEM SECURITY

Number of Lecture hours per week: 5Hrs
Number of credits for theory: 3

Aim of the course

To provide an understanding of the differences between various forms of computer security, where they arise, and appropriate tools to achieve them.

Course Outline

Unit I: Notion of different types of securities - information security - computer security -security goals, relation between security, confidentiality, integrity, availability and authorization, vulnerabilities - principles of adequate protection. Notions of operating security, database security, program security, network security attacks - threats,vulnerabilities and controls. The kind of problems - interception, interruption, modification,fabrication. Computer criminals - amateurs, crackers, career criminals. Methods of defence- control, hardware controls, software controls, effectiveness of controls.

Unit II: Program security - secure programs - fixing faults, unexpected behaviour, types of flaws. Non-malicious program errors - buffer overflows, incomplete mediation. Viruses and other malicious code - kinds of malicious code, how viruses attach, how viruses gain control,prevention, control example - the brain virus, the internet worm, web bugs. Targeted malicious code - trapdoors, Salami attack. Controls against program threats - development controls, peer reviews, hazard analysis**.**

Unit III: Operating system security - protected objects and methods of protection -
memory address protection - fence, relocation, base/bounds registers, tagged architecture,segmentation, paging. Control of access to general objects - directory, access control

list. File protection mechanism - basics forms of protection, single permissions. Authentication - authentication basics, password, authentication process challenge - response, biometrics.Trusted operating systems - security policies for operating systems, models of security -requirement of security systems, multilevel security, access security, limitations of security systems. Trusted operating system design - elements, security features, assurance, system flaws and assurance methods.

Unit IV: Database Security - security requirements - integrity of database, confidentiality and availability, reliability and integrity, sensitive data, interface, multilevel database,proposals for multilevel security.

Unit V: Administrating security - security planning - contents of a security planning, team members, commitment to a security plan, business continuity plans. Risk analysis -the nature of risk, steps of risk analysis. Arguments for and against risk analysis, organizational security policies - purpose and goals of organizational security. Audience, characteristics of a good security policy. Nature of security policies - data sensitivity policy, government agency IT security policy. Physical security - natural disaster, human vandals, interception of sensitive information.

References:

1. C. P. Pfleeger and S. L. Pfleeger, Security in Computing, 4th Edition, Pearson India, ISBN: 9788131727256.
2. Matt Bishop, Computer Security: Art & Science, 1st Edition, Pearson, ISBN: 0201440997.
3. William Stallings, Cryptography and Network Security: Principles and Practice, 6th Edition, Pearson India, ISBN: 9332518777.
4. Michael E. Whitman and Herbert J. Mattord, Principles of Information Security, 4th Edition, Ceneage Learning India Pvt Ltd, ISBN: 8131516458.

Curriculum Plan

Unit/ hours (time required)	**Topics to be taught (input)**	**Procedure (process) Student centric Method of teaching**	**Activity**	**Learning outcome (output)**	**Assessment**
Unit I	Notion of different types of securities - information security - computer security -security goals, relation between security, confidentiality, inte	• Lecture using ICT tools.	Peer group discussion on different types of securities.	To Understand • Concept of different types of securities.	• Assignment • MCQ evaluation

	grity, availability and authorization, vulnerabilities - principles of adequate protection. Notions of operating security, database security, program security, network security attacks - threats, vulnerabilities and controls. The kind of problems - interception, interruption, modification, fabrication. Computer criminals - amateurs, crackers, career criminals. Methods of defence - control, hardware controls, software controls, effectiveness of controls.				
Unit II	**s.**Program security - secure programs - fixing faults, unexpected behaviour, types of flaws. Non-malicious program errors - buffer overflows, incomplete mediation. Viruses and other malicious code - kinds of malicious code, how viruses attach, how viruses gain	• Lecture using ICT Tools • Illustrations • Hands on sessions	• Seminar • Group study	To Understand • Flaws • Viruses • Buffer overflows • Program threats and Controls.	• Q&A exams.

	control,prevention, control example - the brain virus, the internet worm, web bugs. Targeted malicious code - trapdoors, Salami attack. Controls against program threats - development controls, peer reviews, hazard analysis.				
Unit III	Operating system security - protected objects and methods of protection - memory address protection - fence, relocation, base/bounds registers, tagged architecture,segmentation, paging. Control of access to general objects - directory, access control list. File protection mechanism - basics forms of protection, single permissions. Authentication - authentication basics, password, authentication process challenge - response, biometrics.Trusted operating systems - security policies for operating systems, models of security - requirement of	• Lecture using ICT tools • Illustrations	• Hands on sessions • Illustrations	To Understand • Memory address protection • Trusted OS	• Q&A exams

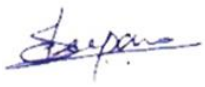
	security systems, multilevel security, access security, limitations of security systems. Trusted operating system design - elements, security features, assurance, system flaws and assurance methods.				
Unit IV	Database Security - security requirements - integrity of database, confidentiality and availability, reliability and integrity, sensitive data, interface, multilevel database,proposals for multilevel security.	• Lecture using ICT tools • Illustrations • Hands on Session	• Hands on sessions • Illustrations	To understand • Database securities	• Lab tests • Q&A exams
Unit V	Administrating security - security planning - contents of a security planning, team members, commitment to a security plan, business continuity plans. Risk analysis - the nature of risk, steps of risk analysis. Arguments for and against risk analysis, organizational security policies - purpose and goals	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand • Security plan • Good security policy • IT security policy	• Assignments

	of organizational security. Audience, characteristics of a good security policy. Nature of security policies - data sensitivity policy, government agency IT security policy. Physical security - natural disaster, human vandals, interception of sensitive information.				

Course Outcomes:

CO:1	CO1: Summarize different types of securities.
CO:2	CO2: Describe information security.
CO:3	CO3: Discuss biometric based authentication.
CO:4	CO4: Describe COA triad.
CO:5	CO5: Discuss program security in system security.
CO:6	CO6: Trusted OS design.
CO:7	CO7: Discuss threats of virus and other malicious codes.Explain the working and controlling mechanism of viruses.
CO:8	CO8: Describes various types of system security challenges.

Faculty in Charge:



Deepa M

Programme: CSS3E01a – CSS4E04a Digital Image Processing

Number of Lecture hours per week	**5**Hrs
Number of credits for theory	**4**

Aim of the course

To be familiar with processing of the images, recognition of the pattern and their applications.

Course Outline

Unit I: Introduction - digital image representation - fundamental steps in image processing- elements of digital image processing systems - digital image fundamentals - elements of visual perception - a simple image model - sampling and quantization - basic relationship between pixels - image geometry.

Unit II: Image transforms - introduction to Fourier transform - discrete Fourier transform(DFT) - properties DFT- other separable image transforms - Walsh, Hadamard and Discrete Cosine transforms. Hotelling transform.

Unit III: Image enhancement - basic grey level transformation - histogram equalization -image subtraction - image averaging - spatial filtering - smoothing, sharpening filters Laplacian filters. Enhancement in the frequency domain - frequency domain filters smoothing, sharpening filters - homomorphic filtering.

Unit IV:Image restoration - model of Image degradation/restoration process - noise models inverse filtering - least mean square filtering - constrained least mean square filtering. Edge detection - thresholding - region based segmentation - boundary representation.

Unit V: Image compression - fundamental concepts of image compression – compression models - information theoretic perspective. Lossless compression - Huffman coding -arithmetic coding - bit plane coding - run length coding. Lossy compression – transform coding - image compression standards.

References:

1. Richard E Woods and Rafael C Gonzalez, Digital Image Processing, 3rd Edition, Pearson Education Singapore Pte Ltd, ISBN: 8131726959.
2. B. Chanda and D.D. Majumder, Digital Image Processing and Analysis, 2nd Edition, PHI Learning Pvt Ltd, ISBN: 8120343255.
3. A.K. Jain, Fundamentals of Digital Image Processing, 2nd Edition, PHI Learning Pvt Ltd, ISBN: 8120309294.
4. W.K. Pratt, Digital Image Processing: PIKS Scientific Inside, 4th Edition, John Wiley, ISBN: 0471767778.

5. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing Analysis and Machine Vision, 3rd Edition, Ceneage Learning India Pvt Ltd, ISBN: 8131518833.

Curriculum Plan

Unit/hours (time required)	**Topics to be taught (input)**	**Procedure (process) Student centric Method of teaching**	**Activity**	**Learning outcome (output)**	**Assessment**
Unit I	Introduction - digital image representation - fundamental steps in image processing - elements of digital image processing systems - digital image fundamentals - elements of visual perception - a simple image model - sampling and quantization - basic relationship between pixels - image geometry.	• Lecture using ICT tools.	Peer group discussion on digital image processing systems.	To Understand • Concept of digital image processing systems.	• Assignment • MCQ evaluation
Unit II	Image transforms - introduction to Fourier transform - discrete Fourier transform(DFT) - properties DFT- other separable image transforms - Walsh, Hadamard and Discrete Cosine transforms. Hotelling transform.	• Lecture using ICT Tools • Illustrations • Hands on sessions	• Seminar • Group study	To Understand • DFT • Cosine transformation. • Hotelling transform	• Q&A exams.
Unit III	Image enhancement - basic grey level transformation - histogram	• Lecture using ICT tools • Illustratio	• Hands on sessions	To Understand • Grey level	• Q&A

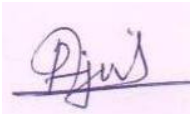
	equalization - image subtraction - image averaging - spatial filtering - smoothing, sharpening filters Laplacian filters. Enhancement in the frequency domain - frequency domain filters smoothing, sharpening filters - homomorphic filtering.	ns	Illustrations	transformation • Filters	exams
Unit IV	Image restoration - model of Image degradation/restoration process - noise models inverse filtering - least mean square filtering - constrained least mean square filtering. Edge detection - thresholding - region based segmentation - boundary representation.	• Lecture using ICT tools • Illustrations • Hands on Session	• Hands on sessions • Illustrations	To understand • Edge detection	• Lab tests • Q&A exams
Unit V	Image compression - fundamental concepts of image compression - compression models - information theoretic perspective. Lossless compression - Huffman coding - arithmetic coding - bit plane coding -	• Lecture using ICT tools • Illustrations • Hands on Session	• Seminar • Group study	To understand • Image compression • Huffman Coding • Compression standards	• Assignments

	run length coding. Lossy compression - transform coding - image compression standards.				

Course Outcomes:

CO:1	CO1: Understand about digital image processing.
CO:2	CO2: Describe Fourier transformation.
CO:3	CO3: Discuss compression methods like Lossless and Lossy compression.
CO:4	CO4: Learn about edge preserving.
CO:5	CO5: Discuss Huffman coding.
CO:6	CO6: Know about Image restoration.
CO:7	CO7: Discuss various filters used for Image processing.
CO:8	CO8: Understand elements of digital image perception.

Faculty in Charge:



Julie P A