E.G.S. PILLAY ENGINEERING COLLEGE

(Autonomous)

Approved by AICTE, New Delhi | Affiliated to Anna University, Chennai

Accredited by NAAC with 'A' Grade | Accredited by NBA (CSE, EEE, MECH)

NAGAPATTINAM – 611 002

MASTER OF COMPUTER APPLICATIONS

Full Time Curriculum and Syllabus

First Year – Second Semester

Course Code	Course Name	L	T	P	C	Maximum Marks		
						CA	ES	Total
Theory Course								
1702CA201	Computer Communication and Networks	3	0	0	3	40	60	100
1702CA202	Operating Systems	3	0	0	3	40	60	100
1702CA203	Software Engineering Methodologies	3	0	0	3	40	60	100
1702CA204	Design and Analysis of Algorithms	3	0	0	3	40	60	100
1702CA205	Object Oriented Programming	3	0	0	3	40	60	100
Laboratory Course								
1702CA206	OOP and Algorithms Laboratory	0	0	4	2	50	50	100
1702CA207	OS and Network Programming Laboratory	0	0	4	2	50	50	100
1704CA208	Life Skill II – Verbal Ability	0	0	2	1	100	-	100

L – Lecture | T – Tutorial | P – Practical | CA – Continuous Assessment | ES – End Semester

1702CA201	COMPUTER COMMUNICATION AND NETWORKS	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To explore various data communication techniques
2. To know network fundamentals and protocols
3. To understand network addressing and routing concepts
4. To understand the requirement of reliable and unreliable communication
5. To understand the functionality and concepts of various application layer protocols

UNIT I DATA COMMUNICATIONS 9 Hours

Data communications and Networking: Communication model, Data transmission concepts and terminology, Transmission media, Data encoding techniques – Digital data communication techniques: Error detection and correction, Line configurations – Multiplexing: FDM, TDM, Statistical TDM.

UNIT II NETWORK FUNDAMENTALS 9 Hours

Network Architecture: The OSI model, TCP/IP model – Network interface layer: Framing – Reliable transmission: stop and wait protocol, sliding window protocols – MAC: Ethernet, Token ring, Wireless LAN, Blue Tooth – Bridges.

UNIT III DATA LINK LAYER 9 Hours

Data link control - Flow Control – Error Detection and Error Correction - MAC – Ethernet, Token ring, Wireless LAN MAC – Blue Tooth - Bridges.

UNIT IV NETWORK LAYER 9 Hours

Network layer functions – circuit switching – packet switching – IP datagram – IPv4 – Sub netting and classless addressing – IPv6 – ARP – Routing protocols: distance vector, link state – ICMP – ICMPv6.

UNIT V TRANSPORT LAYER AND APPLICATION LAYER 9 Hours

Transport Layer**:** Duties of transport layer– User Datagram Protocol – Transmission Control Protocol – Congestion – Congestion control. Application Layer: Application layer Protocols – World Wide Web and HTTP – FTP – Domain name system– Telnet –Electronic mail protocols – SNMP

TOTAL: 45 HOURS

FURTHER READING:

Multimedia in the Internet – real-time interactive protocols – Session Initialization Protocol (SIP) – Peer-to-Peer Paradigm: Chord, Pastry – Transport layer security – Application layer security.

COURSE OUTCOMES:

On the successful completion of the course, students will be able to

CO1: Explain how communication works in data networks and the Internet
CO2: Explain the role of protocols in data networks
CO3: Describe the importance of addressing and naming schemes at various layers of data networks.
CO4: Describe the protocols and services provided by the application layer in the OSI model and describe how this layer operates in sample networks.

REFERENCES:

1. Larry L. Peterson and Bruce S. Davie, "Computer Networks: A Systems Approach", Fifth Edition, Morgan Kaufmann Publishers, 2012.
2. William Stallings, "Data and Computer Communications", Tenth Edition, Pearson, 2013.
3. James F. Kurose, Keith W. Ross, "Computer Networking: A Top-Down Approach", Fifth Edition, Pearson Education, 2012.
4. Forouzan, "Data Communication and Networking", Fifth Edition, TMH, 2012.
5. Andrew S.Tannenbaum and David J. Wetherall, "Computer Networks", Fifth Edition, Pearson Education, 2011.

1702CA202	OPERATING SYSTEMS	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To Learn the Operating System basics.
2. To Study the process management of Operating system .
3. To Gain knowledge in storage management and I/O systems of Operating system
4. To Explore the case studies with various operating systems

UNIT I OPERATING SYSTEMS OVERVIEW 9 Hours

Operating system -Types of Computer Systems - Computer-system operation – I/O structure –Hardware Protection - System components – System calls – System programs – System structure-– Process concept – Process scheduling – Operations on processes – Cooperating processes – Inter-process communication – Communication in client-server systems - Threads-Multithreading Models-Thread Libraries-Threading Issues.

UNIT II PROCESS MANAGEMENT 9 Hours

Scheduling criteria – Scheduling algorithms – Multiple-processor scheduling – Real time scheduling –Thread Scheduling-– Process Scheduling Models - The critical- section problem – Synchronization hardware – Semaphores – Classic problems of Synchronization – Critical regions – Monitors-Usage-Dining philosopher solution using monitor -Deadlock – Deadlock characterization – Methods for handling deadlocks – Recovery from deadlock

UNIT III STORAGE MANAGEMENT 9 Hours

Memory Management – Swapping – Contiguous memory allocation – Paging – Segmentation – Segmentation with paging- Virtual Memory - Background – Demand paging – Process creation – Page replacement – Allocation of frames – Thrashing.

UNIT IV FILE AND I/O SYSTEMS 9 Hours

File concept – Access methods – Directory structure – File-system mounting – Protection - Directory implementation – Allocation methods – Free-space management -Secondary Storage Structure-Mass Storage-Disk Storage and Attachment- Disk scheduling – Disk management – Swap-space management

UNIT V CASE STUDY 9 Hours

The Linux System - History – Design Principles – Kernel Modules – Process Management – Scheduling – Memory management – File systems – Input and Output – Inter-process Communication – Network Structure – Security – Windows 7 - History – Design Principles – System Components – Environmental subsystems – File system – Networking.

TOTAL: 45 HOURS

FURTHER READING:

The course doesn't cover the modern operating system like Mobile and Embedded OS and hence it will be provided us additional content.

COURSE OUTCOMES:

On the successful completion of the course, students will be able to

CO1: Explain the basics of OS.
CO2: Able to demonstrate the mapping between the physical memory and virtual memory
CO3: Able to understand the operating system components and services with the recent OS
CO4: Able to understand file handling concepts in OS perspective

REFERENCES:

1. Abraham Silberschatz, Peter B. Galvin and Greg Gagne, "Operating System Concepts", Ninth Edition, John Wiley and Sons Inc, 2012.
2. Andrew S. Tanenbaum, "Modern Operating Systems", Second Edition, Addison Wesley, 2001..
3. Gary Nutt, "Operating Systems", Second Edition, Addison Wesley, 2001.
4. H M Deital, P J Deital and D R Choffnes, "Operating Systems", Pearson Education, 2004.
5. Andrew S.Tanenbaum "Operating Systems Design and implementation"Third edition Prentice hall, 2006
6. William Stallings "Operating Systems: Internals and Design Principles" 7th Edition, Prentice

1702CA203 SOFTWARE ENGINEERING METHODOLOGIES

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

1. To provide an insight into the processes of software development.
2. To understand and practice the various fields such as analysis, design, development, testing of Software Engineering.
3. To develop skills to construct software of high quality with high reliability.
4. To apply metrics and testing techniques to evaluate the software.

UNIT I INTRODUCTION **9 Hours**

Software Engineering – Product and process – process models - Waterfall Life cycle model – Spiral Model – Prototype Model – fourth Generation Techniques – Agile methods.

UNIT II REQUIREMENT ANALYSIS **9 Hours**

Software Requirements Analysis and Specification – Software Requirements – Problem Analysis – Requirements Specification – Validation – Metrics – Summary.

UNIT III SOFTWARE DESIGN **9 Hours**

Abstraction – Modularity – Software Architecture – Cohesion – Coupling – Various Design Concepts and notations – Real time and Distributed System Design – Documentation – Dataflow Oriented design – Designing for reuse – Programming standards.

UNIT IV SOFTWARE TESTING **9 Hours**

Coding – Programming Practice – Top-down and Bottom-up - structured programming – Information Hiding – Programming style – Internal Documentation Verification – Code Reading – Static Analysis – Symbolic Execution – Code Inspection or Reviews – Unit Testing – Fundamentals – Functional Testing versus structural Testing Coding.

UNIT V SOFTWARE MAINTENANCE AND SOFTWARE METRICS **9 Hours**

Need for Software maintenance – Maintenance models - SCM – Version Control – SCM process – Software Configuration Items – Taxonomy – Basics of Case tools - Scope of Software Metrics – Classification of metrics – Measuring Process and Product attributes – Direct and Indirect measures – Reliability – Software Quality Assurance – Standards.

TOTAL: 45 HOURS

FURTHER READING:

Case Study for Project Plan and SRS, Design of any Application Project, Testing Techniques, COCOMO model, Web Engineering

COURSE OUTCOMES:

On the successful completion of the course, students will be able to

CO1: Understand the basic concepts of various models in software engineering.
CO2: Model the software projects into high level design using DFD,UML diagrams.
CO3: Evaluate the system with various testing techniques and strategies
CO4: Apply various software metrics on software quality products.

REFERENCES:

1. Pankaj Jalote, "An Integrated Approach to Software Engineering", Third Edition, Narosa Publications, 2011.
2. Ian Sommerville, "Software engineering", Ninth Edition, Pearson Education Asia, 2010.
3. Roger S. Pressman, "Software Engineering – A Practitioner‟s Approach", Seventh Edition, Tata
4. McGraw-Hill International Edition, 2009.

1702CA204	DESIGN AND ANALYSIS OF ALGORITHMS	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To understand the problem solving process and writing algorithms
2. To use algorithm design paradigms for algorithm design
3. To analyze the algorithms for time/space complexity

UNIT I ALGORITHM ANALYSIS **9 Hours**

The Role of Algorithms in Computing – Insertion sort – Analyzing algorithms – Designing Algorithms – Growth of Functions – Asymptotic Notation – Standard Notations and Common Functions

UNIT II HEAP SORT AND QUICK SORT **9 Hours**

Heap Sort: Heaps – Maintaining the Heap Property – Building a Heap – Heap Sort Algorithm –Priority Queues – Quick Sort: Description – Performance – Randomized Version – Analysis

UNIT III DESIGN AND ANALYSIS TECHNIQUES **9 Hours**

Introduction to Dynamic Programming – Matrix Chain Multiplication – Longest Common Subsequence – Greedy Algorithms – Activity Selection Problem – Huffman Codes

UNIT IV GRAPH ALGORITHMS **9 Hours**

Representation of Graphs – Representing Attributes – Breadth-First Search – Breadth-First Trees – Depth-First Search – Topological Sort – Strongly Connected Components – Minimum Spanning Trees: Growing a Minimum Spanning Trees – Algorithms of Kruskal and Prim – Single Source Shortest Path – Bellman-Ford Algorithm – Single Source Shortest Path in Directed Acyclic Graphs – Dijkstra‟s Algorithm

UNIT V NP PROBLEMS **9 Hours**

Polynomial Time – Polynomial-time Verification – NP - completeness and Reducibility – NP- Completeness Proofs – NP-complete Problems – Approximation Algorithms – Vertex Cover Problem-Clique decision problem-Node Cover-Chromatic number decision Problem.

TOTAL: 45 HOURS

FURTHER READING:

Backtracking, Branch and Bound Method

COURSE OUTCOMES:

On the successful completion of the course, students will be able to

CO1: Analyze the algorithms for time/space complexity.
CO2: Implement heap sort and quick sort.
CO3: Design algorithms using dynamic programming and Greedy approaches and graph structure to solve real-life problems.
CO4: Analyze problems in terms of polynomial time.

REFERENCES:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford S t e i n , “Introduction to Algorithms", Third Edition, PHI Learning, 2002.
2. Aho, Hopcroft and Ullman, “Data Structures and Algorithms”, Pearson Education, 1983.
3. Anany Levitin, “Introduction to Design and Analysis of Algorithms”, Third, Pearson Education, 2012.
4. Robert Sedgewick and Kevin Wayne, “Algorithms”, Fourth Edition, Pearson Education, 2011.
5. S.Sridhar, “Design and Analysis of Algorithms”, First Edition, Oxford University Press, 2014.

1702CA205	OBJECT ORIENTED PROGRAMMING	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To understand the OO paradigm.
2. To be aware of the OO design technique.
3. To learn the syntax of C++.
4. To be exposed to the file processing and exception handling techniques of C++.
5. To be familiarized with the Standard Template Library.

UNIT I INTRODUCTION **9 Hours**

Introduction - Object-oriented Design - Object-oriented Features- –Types and Declaration –Pointers –Arrays - Structures–Expressions and Statements –Functions –Namespaces and Exceptions.

UNIT II ABSTRACT DATA TYPES **9 Hours**

Classes – Constructors – Destructors – Function Overloading - Operator Overloading – Conversions.

UNIT III INHERITANCE **9 Hours**

Derived Class – Virtual Functions –Polymorphism - Abstract Base Class – Multiple Inheritances.

UNIT IV GENERIC PROGRAMMING **9 Hours**

Templates – Generic Functions – Generic Classes –Standard Template Library –Containers – Iterators Function Objects –Allocators.

UNIT V I/O AND EXCEPTION HANDLING **9 Hours**

Streams - Ostream – Istream –Files –Throwing Exceptions – Try Blocks – Handlers- OOP using C++

TOTAL: 45 HOURS

FURTHER READING:

Java Serialization, RTTI Templates, STL Algorithms and Function Adaptors.

COURSE OUTCOMES:

On the successful completion of the course, students will be able to

CO1: Design and implement C++ programs for any given problem using Class, Polymorphism and Inheritance.

CO2: Write generic programs using STL.

CO3: Write C++ programs using file processing and exception handling techniques.

REFERENCES:

1. Ira Pohl, “Object - Oriented Programming using C++”, Second Edition, Pearson Education, 2003.
2. Bjarne Stroustrup, “The C++ Programming Language”, Fourth Edition, Addison Wesley, 2013.
3. Herbert Schildt, “C++: The Complete Reference”, Fifth Edition, McGraw Hill, 2015.
4. Stanley B. Lippman, Josée Lajoie , C++ Primer, Fifth Edition , Barbara E. Moo ,Pearson Education, 2012.

1702CA206	OOP AND ALGORITHMS LABORATORY	L	T	P	C
		0	0	4	2

COURSE OBJECTIVES:

1. To develop skills in object oriented programming.
2. To learn and implement linear, non linear and tree data structures.
3. To learn Set ADT and Graph data structures and its applications.
4. To study, implement and analyze the different sorting techniques

LIST OF EXPERIMENTS:

1. Classes, Constructors, Destructors, Function overloading and Operator overloading.
2. Implementation of Derived classes, Abstract classes, Multiple inheritance, Static and Dynamic Polymorphism.
3. Implementation of IO Stream classes and Exception Handling.
4. Working with function template and class template.
5. Usage of standard template library and implementation of container and iterators.
6. Divide and Conquer Strategy – Merge Sort and Quick Sort Analysis.
7. Graph Implementation – Traversals
8. Shortest path using Dijkstra‟s Algorithm
9. Minimum Spanning Tree using Prim’s Algorithm.
10. Matrix Chain Multiplication and LCS.
11. Activity Selection and Huffman Coding.

TOTAL: 60 HOURS

COURSE OUTCOMES:

On the successful completion of the course, students will be able to

1. Develop programs in object oriented paradigm.
2. Develop applications using various design techniques
3. Design and implement various graph algorithms.

1702CA207	OS AND NETWORK PROGRAMMING LABORATORY	L	T	P	C
		0	0	4	2

COURSE OBJECTIVES:

1. To know about various Process scheduling algorithms
2. To identify tools used for discovering a computer's network configuration with various operating systems.
3. To learn Socket programming.
4. To study about Networking Tools

LIST OF EXPERIMENTS:

1. Establishing LAN with switch or hub and assigning IP addresses/subnet mask
2. Debugging LAN with ping
3. Client-server programming (Sockets)
4. Network analyzer – Wireshark /Ethereal Tool
5. Traffic Analysis –Tool
6. Protocol Analysis – Tool
7. Implementation of Process scheduling algorithms (FCFS)
8. Implementation of Process scheduling algorithms (SJF)
9. Implementation of Process scheduling algorithms (Round Robin)
10. Implementation of Process scheduling algorithms (Priority)
11. Implementation of deadlock detection algorithm
12. Implementation of deadlock prevention and recovery process
13. Implementation of deadlock avoidance algorithm

TOTAL: 60 HOURS

COURSE OUTCOMES:

On the successful completion of the course, students will be able to

1. Able to Implement process scheduling and deadlock detection, prevention algorithms
2. Able to develop networking applications
3. Able to write Socket programs with TCP/UDP