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, IT, CIVIL, ECE)

NAGAPATTINAM – 611 002

B.TECH INFORMATION TECHNOLOGY

Third Year – Fifth Semester

Course Code	Course Name	L	T	P	C	Maximum Marks		
						CA	ES	Total
Theory Course								
1702IT501	Object Oriented Analysis and Design	3	0	0	3	40	60	100
1702IT502	Distributed Computing	3	0	0	3	40	60	100
1702IT503	Computer Graphics and Multimedia	3	0	2	4	50	50	100
1702IT504	Web Programming	3	0	2	4	50	50	100
	Professional Elective - I	3	0	0	3	40	60	100
	Professional Elective - II	3	0	0	3	40	60	100
Laboratory Course								
1704IT551	Case Tools (Mini Project I)	0	0	2	1	50	50	100
1704GE551	Life Skills: Aptitude I	0	0	2	1	100	-	100
1704IT552	Technical Seminar	0	0	2	1	100	-	100
Total		**18**	**0**	**10**	**23**	**510**	**390**	**900**

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

1702IT501	OBJECT ORIENTED ANALYSIS AND DESIGN	L	T	P	C
		3	0	0	3

PREREQUISITE:

1. Software Engineering and Project Management.

COURSE OBJECTIVES:

1. Learn the basics of OO analysis and design skills.
2. Learn the UML design diagrams.
3. Learn to map design to code.
4. Be exposed to the various testing techniques.

UNIT I UML DIAGRAMS 9Hours

Introduction to OOAD – Unified Process - UML diagrams – Use Case – Class Diagrams– Interaction Diagrams – State Diagrams – Activity Diagrams – Package, component and Deployment Diagrams

UNIT II DESIGN PATTERNS 9 Hours

GRASP: Designing objects with responsibilities – Creator – Information expert – Low Coupling – High Cohesion – Controller - Design Patterns – creational - factory method - structural – Bridge – Adapter - behavioral – Strategy – observer

UNIT III CASE STUDY 9 Hours

Case study – the Next Gen POS system, Inception -Use case Modeling - Relating Use cases –include, extend and generalization - Elaboration - Domain Models - Finding conceptual classes and description classes – Associations – Attributes – Domain model refinement – Finding conceptual class Hierarchies - Aggregation and Composition

UNIT IV APPLYING DESIGN PATTERNS 9 Hours

System sequence diagrams - Relationship between sequence diagrams and use cases Logical architecture and UML package diagram – Logical architecture refinement - UML class diagrams – UML interaction diagrams - Applying GoF design patterns

UNIT V CODING AND TESTING 9 Hours

Mapping design to code – Testing: Issues in OO Testing – Class Testing – OO Integration Testing –GUI Testing – OO System Testing.

TOTAL: 45 HOURS

FURTHER READING / CONTENT BEYOND SYLLABUS / SEMINAR :

1. Software Development.
2. Software Design.

Course Outcomes:

At the end of this course, students will be able to,

CO1: Design and implement projects using OO concepts
CO2: Use the UML analysis and design diagrams
CO3: Apply appropriate design patterns
CO4: Create code from design
CO5: Compare and contrast various testing techniques

REFERENCES:

1. Craig Larman, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development", Third Edition, Pearson Education, 2016.
2. Simon Bennett, Steve Mc Robb and Ray Farmer, "Object Oriented Systems Analysis and Design Using UML", Fourth Edition, Mc-Graw Hill Education, 2012.
3. Erich Gamma, a n d Richard Helm, Ralph Johnson, John Vlissides, **"**Design patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley, 2012.
4. Martin Fowler, "UML Distilled: A Brief Guide to the Standard Object Modeling Language", Third edition, Addison Wesley, 2013.
5. Paul C. Jorgensen, "Software Testing:- A Craftsman‟s Approach", Third Edition, Auerbach Publications, Taylor and Francis Group, 2010.
6. http://nptel.ac.in/

1702IT502	DISTRIBUTED COMPUTING	L	T	P	C
		3	0	0	3

PREREQUISITE:

1. Computer Networks

COURSE OBJECTIVES:

1. To provide knowledge on principles underlying the design of distributed systems
2. To lay the foundations of Distributed Systems.
3. To introduce the idea of Distributed Architecture.
4. To introduce the idea of Distributed operating system and related issues.

UNIT I BASIC CONCEPTS 9 Hours

Characterization of Distributed Systems – Examples – Resource Sharing and the Web – Challenges – System Models– Architectural and Fundamental Models – Networking and Internetworking – Types of Networks – Network Principles- Internet Protocols

UNIT II INTERPROCESS COMMUNICATION AND DISTRIBUTED OBJECTS 9 Hours

Interprocess Communication – The API for the Internet Protocols – External Data Representation and Marshalling –Client –Server Communication – Group Communication – Case Study – Distributed Objects and Remote Invocation– Communication Between Distributed Objects – Remote Procedure Call – Events and Notifications.

UNIT III DISTRIBUTED TRANSACTIONS AND CONCURRENCY CONTROL 9 Hours

Transactions - Locks - Optimistic Concurrency Control - Timestamp Ordering - Comparison - Flat and Nested Distributed Transactions - Atomic Commit Protocols - Concurrency Control in Distributed Transactions – Distributed Deadlocks - Transaction Recovery

UNIT IV RESOURCE MANAGEMENT 9 Hours

Time and Global States-Introduction-Clocks, Events and Process states-Synchronizing physical clocks-Logical time and logical clocks-Global states-Distributed debugging-Coordination and Agreement-Introduction-Distributed mutual exclusion-Elections Algorithm- Multicast communication-Consensus and related problems

UNIT V DISTRIBUTED FILE SYSTEM AND NAME SERVICES 9 Hours

Distributed File Systems-Introduction-File service architecture-Network File System- Name Services – introduction -Name Services and the Domain Name System-Directory Services.

TOTAL: 45 HOURS

FURTHER READING / CONTENT BEYOND SYLLABUS / SEMINAR :

1. Cloud Computing.
2. Service Oriented Architecture.
3. Deep Learning.

Course Outcomes:

At the end of this course, students will be able to,

CO1: Articulate the principles and standard practices underlying the design of distributed systems.
CO2: Explain the core issues of distributed systems.
CO3: Appreciate the difficulties in implementing basic communication in distributed systems.
CO4: Have knowledge on the substantial difficulty in designing distributed algorithms in comparison to centralized algorithms.
CO5: Appreciate the issues in distributed operating system, resource management and distributed file system.

REFERENCES:

1. George Coulouris, Jean Dollimore, Tim Kindberg, ―Distributed Systems Concepts and Design‖, Seventh Edition, Pearson Education Asia, 2016.

2. Introduction to Parallel Computing, Second Edition, AnanthGrama, Anshul Gupta, George arypis, Vipin Kumar,: Addison Wesley 2013

3. Ajay D. Kshemkalyani and MukeshSinghal, ―Distributed Computing: Principles, Algorithms and Systems‖, Cambridge Press. 2014

4. A.S.Tanenbaum, M.Van Steen, ―Distributed Systems‖, Pearson Education, 2012.

5. M.L.Liu, ―Distributed Computing Principles and Applications‖, Pearson Addison Wesley, 2014.

6. Tom White, ―Hadoop: The Definitive Guide‖, O'REILLY Media, 2011.

7. http://nptel.ac.in/

1702IT503	COMPUTER GRAPHICS AND MULTIMEDIA	L	T	P	C
		3	0	2	4

PREREQUISITE:

Engineering Graphics, Computer Programming

COURSE OBJECTIVES:

1. Provide in-depth knowledge of display systems, image synthesis and shapes.
2. Understand basic concepts related to Multimedia including data standards, algorithms and software.
3. Develop multimedia applications by utilizing existing libraries.

UNIT I GRAPHICS FUNDAMENTALS 9 Hours

Introduction-Line Circle and Ellipse Drawing Algorithm-Attribute-Two dimensional geometric transformation-Two dimensional Clipping and Viewing

UNIT II TWO DIMENSIONAL GRAPHICS 9 Hours

Two dimensional geometric transformations - Matrix representations and homogeneous co ordinates, composite transformations: window to-viewport coordinate transformation, Two dimensional viewing functions: clipping operations-Point Clipping - Line Clipping: Cohen Sutherland, Liang Barsky -Polygon Clipping: Sutherland Hodgeman

UNIT III THREE DIMENSIONAL GRAPHICS 9 Hours

3D concepts and object representation:3D display methods, polygon surfaces, tables, equations, meshes, curved lies and surfaces, quadric surfaces, spline representation, cubic spline interpolation methods, Bezier curves and surfaces, B-spline curves and surfaces.

3D transformation and viewing:3D scaling, rotation and translation, composite transformation, viewing pipeline and coordinates, parallel and perspective transformation, view volume and general (parallel and perspective) projection transformations.

UNIT IV ANIMATION 9 Hours

Text: Using Text in Multimedia, computer and text, Font Editing and design tools, hypermedia and hypertext - Image: Making Still Images, color, Image File format, Principles of Animation, animation by computer, making animation - Video: Digital video containers, shooting and editing video.

UNIT V MULTIMEDIA 9 Hours

Basic software tools - Text, image, and sound editing tools - painting and drawing tools, animation tools - making instant multimedia - Office suite - Multimedia authoring tools: Types and page based authoring tools, icon and time based authoring tools.

List of Experiments: 15 Hours

1. Implementation of Line, Circle, Ellipse drawing Using DDA Algorithm and Bresenham Algorithm
2. Implementation of 2D Transformations
3. Implementation of 3D Transformations
4. Implementation of Line Clipping Algorithm
5. Use of various Photo editing tool to solve real time problems and apply various effects
6. Use of various Animation tools to solve real time problems and apply various effects
7. To perform a morphing effect of crying face to sad face to happy face and last to most happiest face.
8. Use of Open GL tool to perform Animation and Virtual Reality effects.

TOTAL: 60 HOURS

FURTHER READING / CONTENT BEYOND SYLLABUS / SEMINAR :

1. Computer Vision.
2. Visualization Techniques.

Course Outcomes:

At the end of this course, students will be able to,

CO1: Apply 2D graphics and algorithms to real world applications

CO2: Create interactive graphics applications using 3D modeling and transformation techniques

CO3: Understand the processes involved in the development of a multimedia product from client brief through to delivery

CO4: Plan and create a multimedia product that includes animation, audio and video

REFERENCES:

1. J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes, Computer Graphics; Principles and practice; Second Edition in C;; Addison Wesley, 2016

2. Computer Graphics - C version; D. Hearn and M. P. Baker; Pearson Education, 2014.
3. Computer Graphics - OpenGL version; D. Hearn and M. P. Baker; Pearson Education, 2015
4. K. Andleigh, KiranThakrar , Multimedia Systems Design, PHI, 2012
5. ZeNian Li, S. Drew, "Fundamentals of Multimedia", PHI, 2012.
6. Donald Hearn and M Pauline Baker, Computer Graphics, Pearson Education, 2nd Edition, 2013.
7. http://nptel.ac.in/

1702IT504	WEB PROGRAMMING	L	T	P	C
		3	0	2	4

PREREQUISITE:

1. Programming in Java Programming.
2. Database Management Systems

COURSE OBJECTIVES:

1. To understand the concept of client / server programming
2. To apply web programming languages for developing web applications
3. To know the unique features of scripting languages

UNIT I WEB ESSENTIALS **9 Hours**

Internet – Web clients – Web servers – Markup languages – Introduction to XHTML-Editing XHTML-Headings-Linking –Tables-Images-Forms-Internal linking – Frames - Lists- Cascading Style Sheets (CSS): Features-Style rule cascading and inheritance - Text properties –CSS box model.

UNIT II CLIENT SIDE PROGRAMMING **9 Hours**

Client side vs. Server side programming languages - Introduction to java script –Control statements I - Control statements II - Functions- Objects – Arrays – PHP Programming

UNIT III SERVER SIDE PROGRAMMING **9 Hours**

Java servlet: Architecture – Servlet life cycle -Simple programs using java servlet– Parameter data – Sessions – Cookies – Other servlet capabilities –Data storage –Servlet and concurrency- JDBC- Connecting a java servlet program to a database

UNIT IV XML AND WEB SERVICES **9 Hours**

XML Namespaces-DTD and XML schema-XML parsers: DOM vs. SAX-XSLT – Xquery - XPath- JSP - Running JSP applications – Java beans classes and JSP - Web services concepts - Web services for clients – WSDL – Representing data types: XML schema – SOAP - J2EE

LIST OF EXPERIMENTS **24 HOURS**

1. Write a html program for Creation of web site with forms, frames, links, tables etc
2. Design a web site using HTML and DHTML. Use Basic text Formatting, Images,
3. Create a script that asks the user for a name, then greets the user with "Hello" and the user name on the page
4. Create a script that collects numbers from a page and then adds them up and prints them to a blank field on the page.
5. Create a script that prompts the user for a number and then counts from 1 to that number displaying only the odd numbers.
6. Create a script that will check the field in Assignment 1 for data and alert the user if it is blank. This script should run from a button.
7. Using CSS for creating web sites
8. Creating simple application to access data base using JDBC Formatting HTML with CSS.
9. Program for manipulating Databases and SQL.
10. Program using PHP database functions.
11. Write a web application that functions as a simple hand calculator, but also keeps a "paper trail" of all your previous work
12. Install Tomcat and use JSP and link it with any of the assignments above
13. Reading and Writing the files using .Net
14. Write a program to implement web service for calculator application
15. Implement RMI concept for building any remote method of your choice.

TOTAL: 60 HOURS

FURTHER READING / CONTENT BEYOND SYLLABUS / SEMINAR :

1. Software Development

2. Mobile Application Development

Course Outcomes

At the end of this course, students will be able to,

CO1: Design web pages using HTML and CSS
CO2: Develop web pages using java script
CO3: Develop server side programming techniques to solve real time application
CO4: Apply database concept to create interactive web pages
CO5: Apply JSP concepts to solve real time applications
CO6: Understand the basic concept of web services

REFERENCES:

1.Jeffrey C Jackson, Web Technology – A computer Science perspective, Person Education, New Delhi, 2016.
2. Frank. P. Coyle, XML, Web Services and the Data Revolution, Addison-Wesley Professional, 2012.
3. Chris Bates, Web Programming – Building Internet Applications, Wiley India, 2013.
4. Deitel, Deitel and Neito, Internet and World wide web – How to program, Pearson education, New Delhi, 2016.
5. Gopalan. N.P, Web Technology A Developer Perspectives, PHI, 2012
6. H.M.Deitel, P.J.Deitel, T.R.Nieto, T.M.Lin, XML How to Program, Pearson Education,2012
7. Ramesh Nagappan, Robert Skoczylas and Rima Patel Sriganesh, Developing Java Web Services, Wiley Publishing Inc., 2011.
8. Steve Graham and Doug Davis, Building Web services with Java, Pearson Education 2011
9.http://nptel.ac.in/
10.http://sololearners.com/
11. http://tutorialpoint.org

1704IT551	CASE TOOLS (MINI PROJECT 1)	L	T	P	C
		0	0	2	1

PREREQUISITE :

Software Engineering and Project Management

COURSE OBJECTIVES:

1. Learn the basics of OO analysis and design skills.
2. Be exposed to the UML design diagrams.
3. Learn to map design to code.
4. Be familiar with the various testing techniques

TO DEVELOP A MINI-PROJECT USING FOLLOWING PROBLEM STATEMENTS

1. Identify Use Cases and develop the Use Case model.
2. Identify the conceptual classes and develop a domain model with UML Class diagram.
3. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams.
4. Draw relevant state charts and activity diagrams.
5. Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.
6. Develop and test the Technical services layer.
7. Develop and test the Domain objects layer.
8. Develop and test the User interface layer.

TOTAL:45 HOURS

REQUIREMENTS:

Argo UML or Eclipse IDE or Rational Suite or Visual Paradigm or equivalent

ADDITIONAL EXPERIMENTS/ INNOVATIVE EXPERIMENTS:

1. Commercial building like sky scrapers
2. Domed structures

COURSE OUTCOMES

At the end of this course, students will be able to,

CO1: Design and implement projects using OO concepts.
CO2: Use the UML analysis and design diagrams.
CO3: Apply appropriate design patterns.
CO4: Create code from design.
CO5: Compare and contrast various testing techniques

REFERENCES:

1. Craig Larman, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development", Third Edition, Pearson Education, 2016.
2. Simon Bennett, Steve Mc Robb and Ray Farmer, "Object Oriented Systems Analysis and Design Using

UML", Fourth Edition, Mc-Graw Hill Education, 2012.
3. Erich Gamma, a n d Richard Helm, Ralph Johnson, John Vlissides, **"**Design patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley, 2012.
4. Martin Fowler, "UML Distilled: A Brief Guide to the Standard Object Modeling Language", Third edition, Addison Wesley, 2013.
5. Paul C. Jorgensen, "Software Testing:- A Craftsman‟s Approach", Third Edition, Auerbach Publications, Taylor and Francis Group, 2010.

1704GE551	LIFE SKILLS: APTITUDE – I	L	T	P	C
		0	0	2	1

PREREQUISITE :

Technical English – I and II

COURSE OBJECTIVES:

1. To brush up problem solving skill and to improve intellectual skill of the students
2. To be able to critically evaluate various real life situations by resorting to Analysis Of key issues and factors
3. To be able to demonstrate various principles involved in solving mathematical problems and thereby reducing the time taken for performing job functions.
4. To enhance analytical ability of students
5. To augment logical and critical thinking of Student

Unit I Introduction to Number System, Basic Shortcuts of addition, Multiplication, Division **6 Hours**

Classification of numbers – Types of Numbers - Divisibility rules - Finding the units digit - Finding remainders in divisions involving higher powers - LCM and HCF Models - Fractions and Digits – Square, Square roots – Cube, Cube roots – Shortcuts of addition, multiplication, Division.

Unit II Ratio and proportion, Averages **6 Hours**

Definition of Ratio - Properties of Ratios - Comparison of Ratios - Problems on Ratios - Compound Ratio - Problems on Proportion, Mean proportional and Continued Proportion Definition of Average - Rules of Average - Problems on Average - Problems on Weighted Average - Finding average using assumed mean method.

Unit III Percentages, Profit And Loss **6 Hours**

Introduction Percentage - Converting a percentage into decimals - Converting a Decimal into a percentage - Percentage equivalent of fractions - Problems on percentages - Problems on Profit and Loss percentage- Relation between Cost Price and Selling price - Discount and Marked Price - Two different articles sold at same Cost Price - Two different articles sold at same Selling Price - Gain% / Loss% on Selling Price.

Unit IV Coding and decoding, Direction sense **6 Hours**

Coding using same set of letters - Coding using different set of letters - Coding into a number - Problems on R-model - Solving problems by drawing the paths - Finding the net distance travelled - Finding the direction - Problems on clocks - Problems on shadows - Problems on direction sense using symbols and notations.

Unit V Number and letter series Number and Letter Analogies, Odd man out **6 Hours**

Difference series - Product series - Squares series - Cubes series - Alternate series - Combination series - Miscellaneous series - Place values of letters - Definition of Analogy - Problems on number analogy - Problems on letter analogy - Problems on verbal analogy - Problems on number Odd man out - Problems on letter Odd man out - Problems on verbal Odd man out

Total: 30 Hours

FURTHER READING / CONTENT BEYOND SYLLABUS / SEMINAR :

Verbal Reasoning, Non-Verbal Reasoning, Quantitative and Qualitative Apptitude

COURSE OUTCOMES:

At the end of the course, the student should be able to

CO1: Learners should be able to understand number and solving problems least time using various shortcut

CO2: Solve problems on averages; compare two quantities using ratio and proportion.

CO3: Calculate concept of percentages, implement business transactions using profit and loss.

CO4: Workout concepts of Coding and Decoding, ability to visualize directions and understand the logic behind a sequence.

CO5: Learners should be able to find a series the logic behind a sequence.

REFERENCES:

1. Arun Sharma, „How to Prepare for Quantitative Aptitude for the CAT", 7th edition, McGraw Hills publication, 2016.
2. Arun Sharma, „How to Prepare for Logical Reasoning for CAT", 4th edition, McGraw Hills publication, 2017.
3. R S Agarwal, „A modern approach to Logical reasoning", revised edition, S.Chand publication, 2017.
4. R S Agarwal, „Quantitative Aptitude for Competitive Examinations" revised edition, S.Chand publication, 2017.
5. Rajesh Verma, "Fast Track Objective Arithmetic", 3rd edition, Arihant publication, 2018.

6. B.S. Sijwalii and InduSijwali, "A New Approach to REASONING Verbal & Non-Verbal", 2nd edition, Arihnat publication, 2014.

1704IT552	TECHNICAL SEMINAR	L	T	P	C
		0	0	2	1

COURSE OBJECTIVES:

1. To develop the self-learning skills to utilize various technical resources available from multiple field.
2. To promote the technical presentation and communication skills.
3. To impart the knowledge on intonation, word and sentence stress for improving communicative competence, identifying and overcoming problem sounds.

The students are expected to make two presentations on advanced topics (recent trends) related to III or IV semester subjects. A faculty guide is to be allotted and he / she will guide and monitor the progress of the student and maintain attendance also. Students are encouraged to use various teaching aids such as power point presentation and demonstrative models.

TOTAL: 45 HOURS

ASSESSMENT PATTERN :

Continuous Assessment (100 Marks)

Distribution of marks for Continuous Assessment	Marks
Presentation I	40
Report	10
Presentation II	40
Report	10
Total	**100**

PROFESSIONAL ELECTIVE – I

1703IT001		ADVANCED DATABASE TECHNOLOGY	L	T	P	C
			3	0	0	3

PREREQUISITE: Data Base Management Systems

COURSE OBJECTIVES:

1. To understand the design of databases
2. To acquire knowledge on parallel and distributed databases and its applications.
3. To study the usage and applications of Object Oriented and Intelligent databases.
4. To understand the emerging databases like Mobile, XML, Cloud and Big Data
5. To implement and maintain an efficient database system using emerging trends.

UNIT I	PARALLEL AND DISTRIBUTED DATABASE	9 Hours

Database System Architectures: Centralized and Client-Server Architectures – Server System Architectures – Parallel Systems- Distributed Systems – Parallel Databases: I/O Parallelism – Design of Parallel Systems Distributed Database Concepts - Distributed Data Storage – Distributed Transactions -Case Studies

UNIT II	INTELLIGENT DATABASE	9 Hours

Active Databases: Syntax and Semantics (Starburst, Oracle, DB2)- Taxonomy Applications- Temporal Databases: Overview of Temporal Databases TSQL2- Deductive Databases-Recursive Queries in SQL- Spatial Databases- Spatial Access Methods- Spatial DB Implementation.

UNIT III	XML DATABASE	9 Hours

XML Databases: XML Data Model – DTD – XML Schema – XML Querying – Web Databases – Open Database Connectivity.

UNIT IV	MULTIMEDIA DATABASE	9 Hours

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

UNIT V	GLOBAL ISSUES	9 Hours

Multidimensional Data Structures – Image Databases – Text / Document Databases – Video Databases – Audio Databases – Multimedia Database Design.

TOTAL: 45 Hours

FURTHER READING: Data Analytics, Data Scientist

COURSE OUTCOMES

At the end of this course, students will able to,

CO1: Develop skills on databases to optimize their performance in practice.

CO2: Analyze each type of databases and its necessity
CO3: Design faster algorithms in solving practical database problems
CO4: Design and implement relational databases, distributed databases, XML databases and multimedia databases.
CO5: Implement the concept of database connectivity with the applications

REFERENCES:

1. Henry F Korth, Abraham Silberschatz, S. Sudharshan, ―Database System Concepts‖, Sixth Edition, McGraw Hill, 2016.
2. R. Elmasri, S.B. Navathe, ―Fundamentals of Database Systems‖, Sixth Edition, Pearson Education/Addison Wesley, 2015.
3. C.J.Date, A.Kannan, S.Swamynathan, ―An Introduction to Database Systems‖, Eighth Edition, Pearson Education, 2012.
4. Carlo Zaniolo, Stefano Ceri, Christos Faloutsos, Richard T.Snodgrass, V.S.Subrahmanian, Roberto Zicari, Advanced Database Systems‖, Morgan Kaufmann publishers,2015
5. Vijay Kumar, ―Mobile Database Systems‖, John Wiley & Sons, 2013.
6. http://nptel.ac.in/.

1703IT002	NETWORK PROGRAMMING AND MANAGEMENT	L	T	P	C
		3	0	0	3

PREREQUISITE: Computer Networks

COURSE OBJECTIVES:

1. To learn the basics of socket programming using TCP Sockets.
2. To learn about Socket Options
3. To learn to develop Macros for including Objects In MIB Structure
4. To have knowledge on SNMPv1and SDN concepts.

UNIT I **SOCKETS AND APPLICATION DEVELOPMENT** **9 Hours**

Introduction to Socket Programming - System Calls - Address conversion functions - POSIX Signal Handling - Server with multiple clients - Boundary conditions - Server process Crashes, Server host Crashes, Server Crashes and reboots, Server Shutdown - I/O Multiplexing - I/O Models -TCP echo client/server with I/O Multiplexing.

UNIT II **SOCKET OPTIONS** **9 Hours**

Socket options – getsockopt and setsock opt functions - Generic socket options - IP socket options - ICMP socket options - TCP socket options - Multiplexing TCP and UDP sockets - SCTP Sockets - SCTP Client/server - Streaming Example - Domain name system - gethostbyname, gethostbyaddr, getservbyname and getservbyport functions - Protocol Independent functions in TCP Client/Server Scenario.

UNIT III **ADVANCED SOCKETS** **9 Hours**

IPv4 and IPv6 interoperability - Threaded servers - Thread creation and termination - TCP echo server using threads - Mutex - Condition variables - Raw sockets - Raw socket creation - Raw socket output - Raw socket input - ping program - trace route program

UNIT IV **SIMPLE NETWORK MANAGEMENT** **9 Hours**

SNMP network management concepts - SNMPv1 - Management information - MIB Structure - Object syntax - Standard MIB‘s - MIB-II Groups - SNMPv1 protocol and Practical issues- Overview of RMON- Statistics and collection- Alarms and Filters..

UNIT V **SOFTWARE DEFINED NETWORK** **9 Hours**

Routes, Paths, And Connections -Traffic Engineering and Control Of Path Selection- Separation Of Data and Control - The SDN Architecture and External Controllers - SDN Across Multiple Devices- Implementing SDN With Conventional Switches- Open Flow Technology and basics- Software Defined Radio (SDR).

TOTAL: 45 Hours

FURTHER READING: Case study of network management

COURSE OUTCOMES

At the end of this course, students will able to,

CO1: Implement Client/Server communications using TCP and UDP Sockets
CO2: Describe the usage of Socket Options for handling various Sockets in programming
CO3: Learn about handling Raw sockets
CO4: Learn the functionalities of SNMP and MIB structure
CO5: Articulate network engineering principles and implementation of SDN.

REFERENCES:

1. W. Richard Stevens, ―UNIX Network Programming Vol-I‖, Fifth Edition, PHI Pearson Education, 2015.
2. William Stallings, ―SNMP, SNMPv2, SNMPv3 and RMON 1 and 2‖, Third Edition, Pearson Edition, 2013.
3. D.E. Comer, ―Internetworking with TCP/IP Vol- I: Sixth Edition, Pearson Edition, 2013.
4. D.E. Comer, ―Internetworking with TCP/IP Vol- III: Client-Server Programming and Application BSD Sockets Version‖, Second Edition, Pearson Edition, 2012.
5. http://nptel.ac.in//

1703IT003	ARTIFICIAL INTELLIGENCE	L	T	P	C
		3	0	0	3

AIM: The main objective of this course is to understand the concepts of Artificial Intelligence and Computer vision.

PREREQUISITE: Computer Networks, Software Engineering and Project Management

COURSE OBJECTIVES:

1. Study the concepts of Artificial Intelligence.
2. Learn the methods of solving problems using Artificial Intelligence.
3. Introduce the concepts of Expert Systems and machine learning.

UNIT I	INTRODUCTION TO AI	9 Hours

Artificial Intelligence – Problem Solving – Production Systems – Algorithms Analysis – Searching Techniques – Case Study: Constraint Statisfication Problem, Hill Climbing

UNIT II	KNOWLEDGE REPRESENTATION	9 Hours

Knowledge Representation – Predicate Calculus – Inference – Forward & Backward Chaining – Bayes Theory – Fuzzy Approach – Case Study: Game Playing

UNIT III	PLANNING	9 Hours

Basic Plan generation – Strips Language – Scheduling - Explanation – Case Study: Graph Coloring, Reactive Systems

UNIT IV	MACHINE LEARNING	9 Hours

Machine Learning Techniques – Types – Approaches – Applications – Case Study: Ontology, Deep Learning

UNIT V	EXPERT SYSTEMS	9 Hours

Expert systems - Architecture of expert systems, Roles of expert systems – Case Study: Recommendation Systems, Smart GRID, Industrial Internet Search Engines, Social Semantics, Natural Language Processing

TOTAL: 45 Hours

FURTHER READING: Machine Vision Systems, Real Time Learning and Decision making systems

COURSE OUTCOMES

At the end of this course, students will able to,

CO1: Understand the basic of Artificial Intelligence and Problem Solving
CO2: Apply various knowledge representation in solve problems
CO3: Explain various planning techniques and case studies
CO4: Understand different machine learning techniques and case studies
CO5: Explain various expert system applications

REFERENCES:

1. Kevin Night and Elaine Rich, Nair B., "Artificial Intelligence (SIE)", McGraw Hill- 2016.
2. Dan W. Patterson, "Introduction to AI and ES", Pearson Education, 2012.
3. Peter Jackson, "Introduction to Expert Systems", 3rd Edition, Pearson Education, 2014.
4. Stuart Russel and Peter Norvig "AI – A Modern Approach", 2nd Edition, Pearson Education 2010.
5. Deepak Khemani "Artificial Intelligence", Tata Mc Graw Hill Education 2013.
6. http://nptel.ac.in/

1703IT004	THEORY OF COMPUTATION	L	T	P	C
		3	0	0	3

AIM: To provide an understanding of the theoretical development of computer science, particularly for finite representations of languages and machines.

PREREQUISITE: Engineering mathematics, Problem Solving Techniques

COURSE OBJECTIVES:

1. Construct finite state machines and the equivalent regular expressions.
2. Prove the equivalence of languages described by finite state machines and regular expressions.
3. Construct pushdown automata and the equivalent context free grammars
4. Construct Turing machines and Post machines.
5. Be aware of Decidability and Un-decidability of various problems.
6. Learn types of grammars

UNIT I	AUTOMATA THEORY	9 Hours

Abstract machines and computation, formal languages and grammars, finite state machines- Deterministic Finite Automata (DFA) - Non-deterministic Finite Automata (NFA) -Equivalence of DFA and NFA- NFA to DFA conversion-Minimization of DFA.

UNIT II	REGULAR EXPRESSIONS AND LANGUAGES	9 Hours

Regular Expression (RE) - Converting Regular Expression to FA- Converting FA to Regular Expression -Proving languages not to be regular – Closure and Decision properties of Regular Expression - Equivalence and minimization of Automata.

UNIT III	CONTEXT FREE GRAMMARS AND PUSH DOWN AUTOMATA	9 Hours
Context-free grammars-Ambiguity in grammars and languages-simplification of content-free grammars, Pushdown automata, deterministic and non-deterministic pushdown automata and their equivalence with context free languages -Chomsky normal form, Greibach normal form- Closure properties of context-free languages.		
UNIT IV	**TURING MACHINES**	**9 Hours**
Turing machines, computable languages and functions, modifications of Turing machines, Multi head and Multi tape Turing Machines - The Halting problem – Partial Solvability – Problems about Turing machine- Chomsky hierarchy of languages.		
UNIT V	**UNSOLVABLE PROBLEMS**	**9 Hours**
Recursive, and recursively enumerable languages; Undecidability, notion of reduction-Undecidable Problems about Turing Machines – Post's Correspondence Problem		
	TOTAL:	**45 Hours**

FURTHER READING: Machine Vision Systems, Real Time Learning and Decision making systems

COURSE OUTCOMES

At the end of this course, students will able to,

CO1: Explain automata theory as the basis of all computer science languages design
CO2: Construct automata for regular expression and perform minimization of automata
CO3: Perform simplification in grammars and build normalized grammars
CO4: Construct Push Down Automata for a simple Application.
CO5: Construct Turing Machine for a simple Application
CO6: Explain Undecidable problems and measure complexity.

REFERENCES:

1. J.E.Hopcroft, R.Motwani and J.D Ullman, ―Introduction to Automata Theory, Languages and Computations‖, Third Edition, Pearson Education, 2016
2. John C.Martin, ―Introduction to Languages and the Theory of Computation‖, Fourth Edition, Tata McGraw Hill, 2012.
3. Kavi Mahesh, ―Theory of Computation, A Problem-solving Approach‖ Wiley India Pvt, Ltd, 2012.
4. H.R.Lewis and C.H.Papadimitriou, ―Elements of The theory of Computation, Second Edition, Pearson Education/PHI, 2013.
5. Peter Linz, "An Introduction to Formal Language and Automata", Third Edition, Narosa Publishers, New Delhi, 2012.
6. Kamala Krithivasan and Rama. R, "Introduction to Formal Languages, Automata Theory and Computation", Pearson Education 2009
7. http://nptel.ac.in

1703IT005	SOFTWARE TESTING METHODS AND TOOLS	L	T	P	C
		3	0	0	3

AIM: The main objective of this course is used to introduce the concepts of software testing & its levels and automated testing tools

PREREQUISITE: Software Engineering and Project Management

COURSE OBJECTIVES:

1. To know the behavior of the testing techniques to detect the errors in the software
2. To understand standard principles to check the occurrence of defects and its removal.
3. To learn the functionality of automated testing tools
4. To understand the models of software reliability

UNIT I	TESTING ENVIRONMENT AND TEST PROCESSES	9 Hours
World-Class Software Testing Model – Building a Software Testing Environment - Overview of Software Testing Process – Organizing for Testing – Developing the Test Plan – Verification Testing – Analysing and Reporting Test Results		
UNIT II	**TESTING TECHNIQUES AND LEVELS OF TESTING**	**9 Hours**
Using White Box Approach to Test design - Static Testing Vs. Structural Testing – Code Functional Testing – Coverage and Control Flow Graphs –Using Black Box Approaches to Test Case Design – Random Testing – Requirements based testing –Decision tables –State-based testing – Cause-effect graphing – Error guessing – Compatibility testing – Levels of Testing - Unit Testing - Integration Testing - Defect Bash Elimination. System Testing - Usability and Accessibility Testing – Configuration Testing - Compatibility Testing - Case study for White box testing and Black box testing techniques		
UNIT III	**INCORPORATING SPECIALIZED TESTING RESPONSIBILITIES**	**9 Hours**
Testing Client/Server Systems – Rapid Application Development Testing – Testing in a Multiplatform Environment – Testing Software System Security - Testing Object-Oriented Software – Object Oriented Testing – Testing Web based systems – Web based system – Web Technology Evolution – Traditional Software and Web based Software – Challenges in Testing for Web-based Software –Testing a Data Warehouse - Case Study for Web Application Testing.		
UNIT IV	**TEST AUTOMATION**	**9 Hours**

Selecting and Installing Software Testing Tools - Software Test Automation – Skills needed for Automation – Scope of Automation – Design and Architecture for Automation – Requirements for a Test Tool – Challenges in Automation – Tracking the Bug – Debugging – Case study using Bug Tracking Tool.

UNIT V	**SOFTWARE TESTING AND QUALITY METRICS**	**9 Hours**

Testing Software System Security - Six-Sigma – TQM - Complexity Metrics and Models – Quality Management Metrics - Availability Metrics - Defect Removal Effectiveness - FMEA - Quality Function Deployment – Taguchi Quality Loss Function – Cost of Quality. Case Study for Complexity and Object Oriented Metrics.

TOTAL:	**45 Hours**

FURTHER READING: Case study of Testing tools like Rational Robot, Amazon Tools

COURSE OUTCOMES

At the end of this course, students will able to,

CO1: Explain the software by applying testing techniques to deliver a product free from bugs
CO2: Evaluate the web applications using bug tracking tools.
CO3: Investigate the scenario and the able to select the proper testing technique
CO4: Explore the test automation concepts and tools
CO5: Deliver quality product to the clients by way of applying standards such as TQM, Six Sigma
CO6: Evaluate the estimation of cost, schedule based on standard metrics

REFERENCES:

1. William Perry, "Effective Methods of Software Testing", Third Edition, Wiley Publishing 2015
2. Srinivasan Desikan and Gopalaswamy Ramesh, "Software Testing – Principles and Practices", Pearson Education, 2014.
3. NareshChauhan, "Software Testing Principles and Practices" Oxford University Press, New Delhi, 2014.
4. Stephen Kan, "Metrics and Models in Software Quality", Addison – Wesley, Second Edition, 2012.
5. LleneBurnstein, " Practical Software Testing", Springer International Edition, Chennai, 2013
6. RenuRajani,Pradeep Oak, "Software Testing – Effective Methods, Tools and Techniques", Tata McGraw Hill, 2010
7. http://nptel.ac.in/

PROFESSIONAL ELECTIVE – II

1703IT006	**MULTICORE PROGRAMMING**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

AIM: This course is provide the advance concepts of process and controllers

PREREQUISITE: Computer Organization and Architecture

COURSE OBJECTIVES:

1. Understand the recent trends in the field of computer architecture and identify performance related parameters
2. Appreciate the need for parallel processing
3. Understand the challenges in parallel and multi-threaded programming
4. To understand the different types of multicore architectures

UNIT I	**INTRODUCTION TO MULTICORE PROCESSORS**	**9 Hours**

Scalable design principles – Principles of processor design – Instruction Level Parallelism, Thread level parallelism. Parallel computer models – Symmetric and distributed shared memory architectures – Multi-core Architectures - Software and hardware multithreading – SMT and CMP architectures – Design issues – Case studies – Intel Multi-core architecture – SUN CMP architecture.

UNIT II	**PARALLEL PROGRAMMING**	**9 Hours**

Performance Issues – Scalability – Synchronization and data sharing – Data races – Synchronization primitives (mutexes, locks, semaphores, barriers) – deadlocks and live locks communication between threads (condition variables, signals, message queues and pipes).

UNIT III	**OPEN MP PROGRAMMING**	**9 Hours**

OpenMP – Threading a loop – Thread overheads – Performance issues – Library functions. Solutions to parallel programming problems – Data races, deadlocks and live locks – Non-blocking algorithms – Memory and cache related issues.

UNIT IV	**MPI PROGRAMMING**	**9 Hours**

MPI Model – MPI constructs – MPI Library –Point-to-point and Collective communication – data decomposition – communicators and topologies – MPI derived data types – Performance evaluation

UNIT V	**MULTITHREADED APPLICATION DEVELOPMENT**	**9 Hours**

Case studies – n-Body solvers – Tree Search – OpenMP and MPI implementations and comparison – Algorithms, program development and performance tuning.

TOTAL:	**45 Hours**

FURTHER READING: Case study of Testing tools like Rational Robot, Amazon Tools

COURSE OUTCOMES

At the end of this course, students will able to,

CO1: Identify the limitations of ILP and the need for multicore architectures
CO2: Discuss the issues related to multiprocessing and suggest solutions
CO3: Solve problems in Parallel Processors
CO4: Explain MPI Programming and topologies
CO5: Develop programs using OpenMP and MPI.

REFERENCES:

1. Shameem Akhter and Jason Roberts, "Multi-core Programming", Intel Press, 2016.
2. Michael J Quinn, Parallel programming in C with MPI and OpenMP, Tata Macgraw Hill, 2013.
3. Peter S. Pacheco, "An Introduction to Parallel Programming", Morgan-Kauffman/Elsevier, 2011.
4. John L. Hennessey and David A. Patterson, " Computer architecture – A quantitative approach", Morgan Kaufmann/Elsevier Publishers, 4th. edition, 2011.
5. David E. Culler, Jaswinder Pal Singh, "Parallel computing architecture : A hardware/ software approach" , Morgan Kaufmann/Elsevier Publishers, 2012.
6. http://nptel.ac.in/

1703IT007	INFORMATION THEORY AND CODING	L	T	P	C
		3	0	0	3

AIM: This course is used to provide various encoding/decoding techniques and multimedia processing techniques

PREREQUISITE: Digital Principles System Design, Computer Networks

COURSE OBJECTIVES:

1. Understand error–control coding.
2. Understand encoding and decoding of digital data streams.
3. Be familiar with the methods for the generation of these codes and their decoding techniques.
4. Be aware of compression and decompression techniques.
5. Learn the concepts of multimedia communication

UNIT I	INFORMATION ENTROPY FUNDAMENTALS	9 Hours

Uncertainty, Information and Entropy – Source coding Theorem – Huffman coding –Shannon Fano coding – Discrete Memory less channels – channel capacity – channel coding Theorem – Channel capacity Theorem

UNIT II	DATA AND VOICE CODING	9 Hours

Differential Pulse code Modulation – Adaptive Differential Pulse Code Modulation – Adaptive subband coding – Delta Modulation – Adaptive Delta Modulation – Coding of speech signal at low bit rates (Vocoders, LPC).

UNIT III	ERROR CONTROL CODING	9 Hours

Linear Block codes – Syndrome Decoding – Minimum distance consideration – cyclic codes – Generator Polynomial – Parity check polynomial – Encoder for cyclic codes – calculation of syndrome – Convolution codes

UNIT IV	COMPRESSION TECHNIQUES	9 Hours

Principles – Text compression – Static Huffman Coding – Dynamic Huffman coding – Arithmetic coding – Image Compression – Graphics Interchange format – Tagged Image File Format – Digitized documents – Introduction to JPEG standards

UNIT V	AUDIO AND VIDEO CODING	9 Hours

Linear Predictive coding – code excited LPC – Perceptual coding, MPEG audio coders – Dolby audio coders – Video compression – Principles – Introduction to H.261 & MPEG Video standards.

TOTAL: 45 Hours

FURTHER READING: Case study of Testing tools like Rational Robot, Amazon Tools

COURSE OUTCOMES

At the end of this course, students will able to,

CO1: Explain various entropy encoding techniques
CO2: Illustrate Data and Voice coding techniques
CO3: Apply error–control method to solve real time problems
CO4: Use compression and decompression techniques to solve real time applications
CO5: Explain audio and video coding

REFERENCES:

1. Simon Haykin, "Communication Systems", 4th Edition, John Wiley and Sons, 2016.
2. Fred Halsall, "Multimedia Communications, Applications Networks Protocols and Standards", Pearson Education, Asia 2013;
3. Mark Nelson, "Data Compression Book", BPB Publication 2013
4. Watkinson J, "Compression in Video and Audio", Focal Press, London, 2012.
5. http://nptel.ac.in/

1703IT008	COMPILER DESIGN	L	T	P	C

		3	**0**	**0**	**3**

AIM: To introduce students to the techniques used in designing and writing compilers

COURSE OBJECTIVES:

1. To enrich the knowledge in various phases of compiler ant its use
2. To extend the knowledge of parser by parsing LL parser and LR parser.
3. To provide practical programming skills necessary for constructing a compiler.
4. Learn to think Java program using real time concepts and paradigms

UNIT I	**INTRODUCTION AND LEXICAL ANALYSIS**	**9 Hours**

Introduction to the translation process, phases of the compiler, compiler tools-Role of lexical analyzer, specification and recognition of tokens-automatic generation of lexical analyzer-The Lexical-Analyzer Generator – Case Study LEX

UNIT II	**SYNTAX ANALYSIS**	**9 Hours**

Introduction – Context-Free Grammars – Writing a Grammar – Top-Down Parsing – Recursive-Descent Parsing and Predictive Parsers - Bottom-up Parsing – Shift-Reduce Parsing - precedence parsing-LR parsers and its Types-Case Study YACC.

UNIT III	**SYNTAX DIRECTED TRANSLATION**	**9 Hours**

Intermediate code generation, interpreters, intermediate languages-syntax trees, postfix code, triples and indirect triples, syntax directed translation of simple statements. Type checking, type systems, type expressions, type conversion and overloading

UNIT IV	**CODE GENERATION**	**9 Hours**

Issues in the Design of a Code Generator – The Target Language – Addresses in the Target Code – Basic Blocks and Flow Graphs – Register allocation and assignment– A Simple Code Generator – Case Study: Peephole Optimization.

UNIT V	**CODE OPTIMIZATION AND RUN TIME ENVIRONMENT**	**9 Hours**

Sources of optimization, optimization of basic blocks, data flow analysis, code improving transformations. Run time environments- storage organization and allocation strategies, Parameter passing, symbol tables.

	TOTAL:	**45 Hours**

FURTHER READING: Case study of Testing tools like Rational Robot, Amazon Tools

COURSE OUTCOMES

At the end of this course, students will able to,

CO1: Describe the various phases of a compiler
CO2: Construct DFA from a given regular expression
CO3: Examine Top-down and Bottom-up parsing Techniques
CO4: Write intermediate code to solve optimization problems
CO5: Identify various types of optimizations on intermediate code and generate assembly code

REFERENCES:

1. Aho, A.V., Lam, M., Sethi, R. and Ullman, J.D., "Compilers: Principles, Techniques and Tools", 2nd Ed., Pearson Education 2014
2. Tremblay, J.P. and Sorenson, P.G., "Theory and Practice of Compiler Writing", SR Publications 2012.
4. Louden, K.C., "Compiler Construction: Principles and Practice", Course Technology, 2012
5. Fischer C.N. and LeBlanc R.J. ―Crafting a Compiler with C‖, Benjamin Cummings, 2012.
6. Bennet J.P., "Introduction to Compiler Techniques‖, Second edition, Tata McGraw-Hill, 2010.
7. http://nptel.ac.in/

1703IT009	**EMBEDDED SYSTEMS**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

AIM: To introduce students to the embedded systems, Real time OS and programming.

COURSE OBJECTIVES:

1. To understand the Basic concepts of Embedded systems
2. To understand the architecture and functions of PIC microcontroller
3. To Learn Assembly language programming
4. To understand the basics of RTOS and to learn the method of designing a real time systems

UNIT I	**INTRODUCTION TO EMBEDDED SYSTEMS**	**9 Hours**

Introduction – Features – Characteristics of Embedding Computing Applications – Concept of Real time Systems – Challenges in Embedded System Design– Microprocessors – ALU– Von Neumann and Harvard Architecture –CISC and RISC – Instruction pipelining. Design Process –Requirements – Specifications –Architecture Design – Designing of Components –System Integration

UNIT II	**PIC MICROCONTROLLER**	**9 Hours**

PIC Microcontroller: Characteristics – Features– Architecture of Microchip PIC16F877– Instruction Set – External Interrupts– Timers– watch-dog timer– I/O port Expansion– analog-to-digital converter–UART– I2C and SPI Bus for Peripheral Chips– Accessories and special features– Case study: Bar-code scanner, Laser printer, Underground tank monitoring.

UNIT III	**REAL – TIME OPERATING SYSTEM**	**9 Hours**

Software architecture and RTOS: Software Architecture: Round Robin – Round Robin with interrupts – Function Queue. Scheduling Architecture: Architecture – Tasks and Task States – Tasks and Data – Semaphores and Shared Data Message Queues –Inter Process Communication – Signal Functions – Socket Programming – Mail Boxes and pipes –Timer Functions – Events – Memory Management – Interrupt Routines – Basic Design of RTOS.

UNIT IV	**RTOS PROGRAMMING**	**9 Hours**

Basic Functions – Types of RTOS – RTOS µCOS – RTLinux – Real Time Linux Functions Programming with RTLinux – Case Study.

UNIT V	**SOFTWARE DEVELOPMENT TOOLS**	**9 Hours**

Development Tool: Cross-Compiler – Cross-Assemblers – Linker/locator – PROM Programmers – ROM Emulator – In-Circuit Emulators – Debugging Techniques – Instruction set simulators– The assert macro – Testing using laboratory tools

TOTAL: 45 Hours

FURTHER READING: Automatic Chocolate Vending Machine, Bio-Metric Machine, etc

COURSE OUTCOMES

At the end of this course, students will able to,

CO1: Able to understand the functionality of PIC microcontroller
CO2: Able to design and control real time control systems
CO3: Able incorporate enhanced features in the embedded systems through software
CO4: Able to rectify minor problems by troubleshooting
CO5: Acquire the knowledge of real time operating system and implement real time functions

REFERENCES:

1. Rajkamal, "Embedded System: Architecture, Programming and Design" Tata Mcgraw - Hill Education, Second Edition, 2016.
2. B.Kanth Rao, "Embedded Systems" PHI Learning Private Limited, 2012.
3. Marilyn Wolf, "Computers as a Component" Third Edition, Morgan Kaufmann Series 2014.
4. A.P.Godse&A.O.Mulani "Embedded Systems" Third Edition, Technical publications 2012.
5. Mohamed Rafiquzzaman, "Microprocessors and Microcomputer-based system design", CRC Press, Second Edition, 2013
6. http://nptel.ac.in/

1703IT010	**MAINFRAME COMPUTING**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

AIM: To introduce students to the embedded systems, Real time OS and programming.

COURSE OBJECTIVES:

1. Learn the concepts of mainframe and usecases
2. Study the concepts of capacity, scalability, Integrity and security
3. Learn accessing and managing mainframe

UNIT I	**NEW MAINFRAME**	**9 Hours**

Mainframe concepts-an evolving architecture- mainframe computer users factors contributing to mainframe use – mainframe workloads

UNIT II	**CAPACITY**	**9 Hours**

Capacity – elements of a system required for capacity – few server Vs Many server – service level agreement – managing the system to the SLA – architecture, running work and capacity – several servers on one physical machine – parallel sysplex and its measurements.

UNIT III	**SCALABILITY, INTEGRITY AND SECURITY**	**9 Hours**

Introduction to scalability – scalability concepts – scalability implementation on IBM system – integrity – security – introduction to availability – Inhibitors to availability - redundancy – z/OS elements for availability – Disaster recovery.

UNIT IV	**ACCESSING LARGE AMOUNT OF DATA**	**9 Hours**

Introduction – channel subsystem – control unit- DASD CKD architecture and DASD subsystem – multiple allegiance/Parallel Access volumes – database and Data sharing – Data placement and management

UNIT V	**SYSTEM MANAGEMENT AND AUTONOMIC COMPUTING**	**9 Hours**

Introduction – system data – configuration management – operating management – performance management – problem management – introduction to autonomic computing – self healing – self protecting – self optimizing.

TOTAL: 45 Hours

FURTHER READING: Automatic Chocolate Vending Machine, Bio-Metric Machine, etc

COURSE OUTCOMES

At the end of this course, students will able to,

CO1: Control and co-operate all acquisition of computers with their associated software and peripherals.
CO2: Maintain a standard configuration on all computers.

CO3: Ensure efficient and effective use of computers by all users throughout Council.
CO4: Creating datasets and performing operations on them
CO5: Identify the different programming languages and Operating systems used on mainframe

REFERENCES:

1. Mike Ebbers,Frank Byrne, Pilar Gonzalez Adrados,Rodney Martin and Jon Veilleux "Redbook – Introduction to Mainframe - Large Scale Commercial Computing". First Edition December 2016, IBM Corp.
2. Lydia Parziale, Edi Lopes Alves, Klaus Egeler, Clive Jordan" Introduction to the New Mainframe: z/VM Basics", IBM Redbooks, 2014
3. http://nptel.ac.in