SEVENTH SEMESTER IT 2018-19 (PATTERN A-15)

COURSE CODE: IT40110 **REF NO: To be filled by CD office**

SOFTWARE PROJECT MANAGEMENT

Credits: 03 **Teaching Scheme:** Theory 03 Hrs/Week

Prerequisites:

1. Software Engineering

Objectives:

- To understand the methods used to evaluate and select projects for investment of funds
- To gain knowledge on the principles and techniques of software project management
- To introduce organization behavior and general management techniques used for project management

Course Details:

Unit I: Project Evaluation (8Hrs)
U1.1 Software Project Definitions

Software Project Categorization, Software VS other projects, Stakeholders, Project Success and Failure, Software project Activities, Practices & Standards, Selecting Process Models (Spiral, Incremental, Prototyping, RAD, Agile), software maintenance model.

U1.2. Estimation & Evaluation Techniques

Business Case & Cost Benefit Analysis, Cash Flow Forecasting, Cost-Benefit Evaluation Techniques (NP, CBR, ROI, NPV, IRR), Risk Analysis for Project Evaluation, Program management, Project effort and cost estimation; Basis of estimation, Estimation method categorization, SLOC, Analogy, Delphi technique, FP Analysis, COCOMO II, Staffing pattern; Norden's & Putnam's work, Schedule compression

U1.3. Self Study
Portfolio management, Program management, Estimation using FP Mark II / COSMIC FP, Capers Jones rules of thumb

Unit 2: Project Planning and Risk Management (8Hrs)

U2.1 Project Planning

Stepwise planning, Identification of Project product and activities (PBS & PFD), Activity based approach (WBS), Sequencing and Scheduling of Activities, Precedence Network Diagram: AON and AOA conventions, Network Planning, CPM technique

U2.2: Risk Management
Nature and Types of Risks, Risk Management framework, Hazard Identification, Hazard Analysis, Risk Planning and Control, Schedule Risk, PERT Technique

U2.3. Self Study
Project Planning Tools (such as GanttProject, LibreProject, MSProject), SPMP documentation, Monte Carlo Simulation techniques, Risk matrix, Measuring Risk Exposure

Unit3: Project Monitoring and Control (8Hrs)

U3.1 Schedule and Cost Monitoring

Collecting Data & Reporting, Graphical Visualization techniques, Cost Monitoring, Earned Value analysis, Requirements management, Change Control.

U3.2 Contract Management:

Types of Contracts, Stages in Contract Placement, Typical Terms of a Contract, Contract Management and Acceptance.

U3.3 Self Study:

Software Configuration Management (SCM), SCM Tools, Project Reviews

Unit 4: Software Quality Management (4 Hrs)

U4.1: Software Quality:

Testing and Software Reliability, Metrics, ISO and CMMI Quality management models, Quality enhancement techniques (Review, Inspection, Formal methods, Best practices and Lessons learnt report)

U4.2 Self Study:

Test automation, Six Sigma, PSP/TSP, Quality Circle

Unit 5: People Management (8 Hrs)

U5.1: Organizational Behavior, Safety and Ethics:

Selecting the right person for a job, Motivation (Taylor's model, expectancy theory), Job characteristic models, Leadership styles, Stress, Health and Safety, Professional Ethics

U5.2: Organization & Project team Structure:

Working in Groups, Decision Making, Organizational structure and Project team structures,

U5.3 Self Study

Maslow's need hierarchy, Herzberg's two factor theory, Safety & Security Standards, Professional Code of Conduct

Text Book

1. Bob Hughes, Mike Cotterell, Rajib Mall, "Software Project Management", 6th Edition, Tata McGraw Hill, 2017.

References:

1. Royce, "Software Project Management", Pearson Education, 1999.
2. Robert K. Wysocki, Effective Software Project Management, Wiley, 2009.

Course Outcomes:

Upon completion of the course the graduate students will be able to

1. Estimate project cost and perform cost-benefit evaluation among projects
2. Perform project scheduling, activity network analysis and risk management
3. Apply schedule and cost control techniques for project monitoring including contract management.

4. Apply quality models in software projects for maintaining software quality and reliability.
5. Use suitable project organization structure, leadership, decision and motivation styles, proper safety and ethical practices and be responsible to the society.

SEVENTH SEMESTER IT 2018-19 (PATTERN A-15)

COURSE CODE: IT40110 **REF NO: To be filled by CD office**

SOFTWARE PROJECT MANAGEMENT - TUTORIAL

Credits: 03 **Teaching Scheme:** Tutorial 01 Hrs/Week

Prerequisites:

1. Software Engineering

Objectives:

- To understand the methods used to evaluate and select projects for investment of funds
- To gain knowledge on the principles and techniques of software project management
- To introduce organization behavior and general management techniques used for project management

Course Details:

List of Contents

Tutorial No. 1: Software project management and its necessity, cost-benefit evaluation techniques.

Tutorial No. 2: Project cost estimation techniques.

Tutorial No. 3: Activity scheduling techniques such as CPM and PERT.

Tutorial No. 4: Risk analysis.

Tutorial No. 5: Cost monitoring.

Tutorial No. 6: Contract management and its necessity.

Tutorial No. 7: Software metrics and process capability models.

Tutorial No. 8: Software quality enhancement techniques.

Tutorial No. 9: People management in software environment.

Tutorial No. 10: Project team structure.

Text Book

1. Bob Hughes, Mike Cotterell, Rajib Mall, "Software Project Management", 6th Edition, Tata McGraw Hill, 2017.

References:

1. Royce, "Software Project Management", Pearson Education, 1999.
2. Robert K. Wysocki, Effective Software Project Management, Wiley, 2009.

Course Outcomes:

Upon completion of the course the graduate students will be able to

1. Estimate project cost and perform cost-benefit evaluation among projects

2. Perform project scheduling, activity network analysis and risk management
3. Apply schedule and cost control techniques for project monitoring including contract management.
4. Apply quality models in software projects for maintaining software quality and reliability.
5. Use suitable project organization structure, leadership, decision and motivation styles, proper safety and ethical practices and be responsible to the society.

SEVENTH SEMESTER IT 2018-19 (PATTERN A-15)

COURSE CODE: IT40310 **REF NO: To be filled by CD office**

SOFTWARE PROJECT MANAGEMENT-LAB

Credits: 1 **Teaching Scheme: - Laboratory 2 Hrs/ week**

Prerequisites: Computer Programming (C/Java/Python/C++), Microsoft Excel

Objectives:

- To develop Graphical User Interface (GUI) based applications for various estimation techniques.
- To make the students familiar with open source tools for activity scheduling techniques.
- To use open source tool for software configuration management.

Course Details:

1. Implementation of estimation techniques any programming language.
2. Study of ProjectLibre tool for activity scheduling techniques.
3. Study of GanttProject tool for activity scheduling techniques.
4. Study of Git tool for software configuration management.

List of Laboratory Exercises:

Experiment 1: Compute function points and COCOMO using online tool such as Tiny Calculator and STRS COCOMO calculator.

Experiment 2: Implement function points computation using any programming language.

Experiment 3: Implement COCOMO estimation using any programming language.

Experiment 4: Implement Halstead's software science using any programming language.

Experiment 5: Creating Work Breakdown Structure (WBS) using ProjectLibre tool.

Experiment 6: Draw Gantt chart and find critical path using ProjectLibre tool.

Experiment 7: Draw Gantt chart and find critical path using GanttProject tool.

Experiment 8: Draw Gantt chart using Microsoft Excel.

Experiment 9: Implement cost-benefit analysis using any programming language.

Experiment 10: Perform cost-benefit analysis using Microsoft Excel.

Experiment 11: Track different versions of a software using Git tool.

Text Book

1. Bob Hughes, Mike Cotterell, Rajib Mall, "Software Project Management", 6th Edition, Tata McGraw Hill, 2017.

References:

1. Royce, "Software Project Management", Pearson Education, 1999.
2. Robert K. Wysocki, Effective Software Project Management, Wiley, 2009.
3. Tiny tool for Function point and COCOMO calculation. http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/gamel/cocomo.html
4. STRS COCOMO calculation. https://strs.grc.nasa.gov/repository/forms/cocomo-calculation/
5. ProjectLibre. https://sourceforge.net/projects/projectlibre/
6. GanttProject. http://www.ganttproject.biz/download
7. Git. https://git-scm.com/download/win
8. Pro Git Book. https://git-scm.com/book/en/v2

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Use programming languages to develop estimation based softwares.
2. Identify the usage of various activity scheduling tools.
3. Identify the usage of software configuration management tool such as Git.
4. Use Microsoft Excel to perform cost benefit analysis and activity scheduling.

COURSE CODE: IT40106 REF NO: To be filled by CD office

Software Engineering (SE)

Credits: 3 **Teaching Scheme: - Theory 3 Hrs/Week**

Prerequisites: Computer Programming, Object-Oriented Concepts, Concepts of Data Modeling

Objectives: This course helps to

- Apply basic principles and techniques of software development in actual problem domain
- Understand theories, methods, and technologies applied for professional software development.
- Model and test real world problems
- Gain knowledge on the emerging software engineering trends

Unit 1: Software Development Life Cycle (10 Hrs)

U1.1 Life Cycle Models: Introduction to Information System and software Engineering, Software Processes, Software Development Life Cycle Models: Waterfall Model, Iterative Waterfall Model, Incremental Models, Evolutionary model, Prototyping Model, Agile process, XP and Scrum
U1.2: Requirement Engineering: Inception, Elicitation, Elaboration, Negotiation, Specification, Validation, management, Software project estimation: LOC and FP

U1.2. Self Study: V Model, DSDM, COCOMO method for estimation

Unit 2 Function-Oriented Analysis & Design (06 Hrs)

U2.1. Function-oriented Design: Introduction to Structured Analysis, Data Flow Diagram, Process Specification, Structured Design Methodologies: Coupling and Cohesion, Structure Chart, Mapping DFD into Structure Chart, Metrics.

U2.2. Self Study: High Level Design Metrics, Component Level Design Metrics

Unit 3 Object-Oriented Analysis & Design (10 Hrs)

U3.1 Modelling Techniques using UML: The Unified Approach to Modelling, Unified Modelling Language (UML) overview, Class and Object diagram, Class relationship: Association, Aggregation, Composition, Generalization, Activity diagram, State diagram, Interaction diagrams, Sequence diagram, Collaboration / Communication Diagram, Component Diagram, and Deployment Diagram.

U3.2 Object-Oriented Analysis & Design: Use-Case Modeling, Use-Case Realization, Types of Classes: Class Classification Approaches: Noun Phrase Approach, CRC Card Approach, Use-case Driven Approach Identification of Classes, Relationship, Attributes and Method, Design Patterns.

U3.3 Self Study: Booch OOD Model, Rumbaugh's OMT model, Jacobson's OOSE model, Timing diagrams, Design Patterns

Unit 4: Implementation and Quality Assurance (06 Hrs)

U4.1. Implementation: Coding standards and guidelines, Human computer interface, Code review and inspection, software maintance.

U4.2. Quality Assurance: Testing Fundamentals, Verification & Validation, Black Box Testing, White Box Testing, Test Case preparation, McCabe's complexity, Unit Testing, Integration Testing, System Testing, Performance Testing. Quality management system.

U4.3. Self Study: Regression Testing, Object-Oriented Integration and System Testing.

Unit 5 Software Architecture trend **(04 Hrs)**

U5.1. Architecture Styles: Repository, Layered, Pipe-Filter, Call-Return, Peer-Peer, Publish-Subscribe, Client-Server, Two-Tier, Three-Tier, N-Tier, and Heterogeneity in Architecture, CORBA, COM/DCOM.

U5.3. Self Study: Service Oriented Architecture (SOA), Software as a Service (SaaS), Model Driven Design (MDD).

Note: Five assignments to be given to the students on self study, comprising of one assignment from each unit.

Course Outcome:

Upon completion of the course, graduates will be able to –

1. Use suitable process model for software development, and carry out requirement engineering process.
2. Carry out Analysis and Design activities using function oriented analysis and design methodologies.
3. Carry out Analysis and Design activities using Object oriented modelling techniques.
4. Use the techniques, skills for software development and software testing.
5. Differentiate different architecture trends.

Text Books:

T1. "Software Engineering: A Practitioner's Approach", Roger S. Pressman, McGraw Hill, 6/e, 2005.
T2. "Fundamentals of Software Engineering", Rajib Mall, PHI, 3rd Edition, 2009.
T3. "Object-Oriented Modeling and Design with UML, Michael R Blaha, James R Rambaugh, Pearson, LPE

Reference Books:

R1. "Software Engineering", Ian Sommerville, Addison-Wesley, 9th Edition, 2010, ISBN- 13: 978-0137035151.
R2. "System Analysis, Design, and Development: Concepts, Principles, and Practices", Charles S. Wasson,, John Wiley & Sons, Inc.,ISBN-13 978-0-471-39333-7, 2006.
R3. "Unified Modeling Language Users Guide", Grady Booch, James Rambaugh, Ivar Jacobson, 2nd Edition, Addison- Wesley, ISBN – 0321267974
R4. "Unified Modeling Language Reference manual", Grady Booch, James Rambaugh, Ivar Jacobson, Pearson India, ISBN – 9788177581614
R5. Scrum Essentials: Agile software Development for Project Managers, Scrum Masters, Product Owners, and Stakeholders", Troy Dimes, Create Space Independent publishing Platform, ISBN: 9781500970512

SEVENTH SEMESTER IT 2018-19 (PATTERN A-15)

COURSE CODE: IT40106 REF NO: To be filled by CD office

SOFTWARE ENGINEERING-TUTORIAL

Credits: 01 **Teaching Scheme:** Tutorial 01 Hrs/Week

Prerequisites: Computer Programming, Object-Oriented Concepts, Concepts of Data Modeling

Objectives: This course helps to

- Apply basic principles and techniques of software development in actual problem domain
- Understand theories, methods, and technologies applied for professional software development.
- Model and test real world problems
- Gain knowledge on the emerging software engineering trends

List of Tutorials

Tutorial No. 1: Develop requirements specification for a given problem (The requirements specification should include both functional and non-functional requirements.

(For a set of about 20 sample problems, see the questions section of Chapter 6 of Fundamentals of Software Engineering Book by Rajib Mall)

Tutorial No. 2: Develop DFD Model (Level 0, Level 1 DFD and Data Dictionary) of the sample problem

Tutorial No. 3: Develop Structured Design for the DFD model developed

Tutorial No. 4: Design test cases using Equivalence class partitioning and Boundary value analysis technique for given problems

Tutorial No. 5: Develop UML Use Case model for a problem

Tutorial No. 6: Develop Class diagrams

Tutorial No. 7: Develop Sequence diagrams from the identified Class diagrams

Tutorial No. 8: Develop State Transition diagram for an object from a given class diagram

Tutorial No. 9: Develop the Activity diagram.

Tutorial No. 10: Analysis of Architecture style

Course Outcome:

Upon completion of the course, graduates will be able to –

1. Use suitable process model for software development, and carry out requirement engineering process.

2. Carry out Analysis and Design activities using function oriented analysis and design methodologies.
3. Carry out Analysis and Design activities using Object oriented modelling techniques.
4. Use the techniques, skills for software development and software testing.
5. Differentiate different architecture trends.

Text Books:

T1. "Software Engineering: A Practitioner's Approach", Roger S. Pressman, McGraw Hill, 6/e, 2005.
T2. "Fundamentals of Software Engineering", Rajib Mall, PHI, 3rd Edition, 2009.
T3. "Object-Oriented Modeling and Design with UML, Michael R Blaha, James R Rambaugh, Pearson, LPE

Reference Books:

R6. "Software Engineering", Ian Sommerville, Addison-Wesley, 9th Edition, 2010, ISBN- 13: 978-0137035151.
R7. "System Analysis, Design, and Development: Concepts, Principles, and Practices", Charles S. Wasson,, John Wiley & Sons, Inc.,ISBN-13 978-0-471-39333-7, 2006.
R8. "Unified Modeling Language Users Guide", Grady Booch, James Rambaugh, Ivar Jacobson, 2nd Edition, Addison- Wesley, ISBN – 0321267974

COURSE CODE: IT40306 **REF NO: To be filled by CD office**

SOFTWARE ENGINEERING-LAB

Credits: 1 **Teaching Scheme: - Laboratory 2 Hrs/week**

Prerequisites: Object Oriented Concepts, Programming in C or Java

Objectives:

- Apply basic principles and techniques of software engineering in actual problem domain
- Develop skills in modelling tools
- Develop design models for software solution to different problems

Course Details:

List of Laboratory Exercises:

Experiment 1: Study of specific problem case, prepare the system requirement specification, and the context diagram.

Experiment 2: Analyze the problem and create the Data Flow Diagram (at least up to 2 levels). Write the data dictionary.

Experiment 3: Develop the module structure chart for the problem.

Experiment 4: Develop UML Use Case model for a problem

Experiment 5: Identify entity objects and classes and develop class diagram. Depict the class relationship using UML.

Experiment 6: To depict the dynamic behaviour using Activity diagram.

Experiment 7: To depict the dynamic behaviour of the target system using sequence diagram.
The Sequence diagram should be based on the Scenarios generated by the inter-object Communication. The model should depict:

a. Discrete, distinguishable entities (class).
b. Controller classes as required
c. Events (Individual stimulus from one object to another)
d. Conditional events and relationship representation.

Experiment 8: Draw the collaboration / Communication diagram

Experiment 9: To depict the state transition with the life history of objects of a given class model. The model should depict:

a. Possible ways the object can respond to events from other objects.
b. Determine of start, end, and transition states.

Experiment 10: Develop test cases for the Java code and execute using JUnit.

(Model drawing tools like Magic Draw &Smart Draw etc. and open source graphics editing tools will be used)

Text Books:

T1. Jim Arlow, Ila Neustadt, "UML 2 and Unified Process: Practical Object Oriented Analysis and Design", 2nd Edition, Addison- Wesley, ISBN – 0321321278.
T2. Tom Pender, "UML Bible", John Wiley & sons, ISBN – 0764526049

Reference Books:

R1. Grady Booch, James Rambaugh, Ivar Jacobson, "Unified Modeling Language Users Guide", 2nd Edition, Addison- Wesley, ISBN – 0321267974
R2. Erich Gamma, Richard Helm, Ralph Johnson,"Design Patterns: Elements of Reusable Object-Oriented Software" (Addison-Wesley Professional Computing Series) ,John Vlissides, Publisher: Addison-Wesley Professional, 1st edition (January 15, 1995) , ISBN-10: 0201633612 ISBN-13: 978-0201633610
R3. Steven Kelly, Juha-Pekka Tolvanen, Domain-Specific Modeling: Enabling Full Code Generation, John Wiley & Sons, Inc., ISBN 978-0-470-03666-2, 2008
R4. Mellor, Scott, Uhl, Weise, "MDA Distilled", Pearson Education, ISBN 81-297-0529X

Note: At least one Text Book and one Reference Book must be from Foreign Author/Foreign Publisher.

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Identify the Requirements from a given problem
2. Design DFD model of a system
3. Develop different UML diagrams
4. Develop test cases for validation of requirement specification and design

COURSE CODE: IT42123 **REF NO: To be filled by CD office**

WIRELESS SENSOR NETWORK

Credits: 03 **Teaching Scheme:** - Theory 3 Hrs/Week

Prerequisites:

1. Data Communication and Computer Networks
2. Computer Programming
3. Design & Analysis of Algorithms
4. Operating System

Objectives:

A wireless sensor network (WSN) is a network of spatially distributed autonomous sensors those monitor physical or environmental conditions and cooperatively pass their data through the network to a main location. This course introduces the wireless sensor networks technology and discusses challenges in the design and management of wireless sensor networks.

Course Details:

Unit 1 Title: Overview of WSN (6 Hrs)

U1.1 Introduction to WSN: Introduction, Challenges of WSN, Advantages of WSN over Ad Hoc wireless Network, overview of sensors, WSN applications - structural health monitoring, traffic control, healthcare, pipeline monitoring, precision agriculture, active volcano, underground mining, Wireless sensor node architecture: hardware and software components of a sensor node. Embedded operating systems structure and execution environments. Programming paradigms and application programming interfaces.

U1.2 Self Study: Sensor Node Hardware – Berkeley Motes, C Motes etc., Programming Challenges, Node-level, software platforms, Node-level Simulators, TinyOS and nesC.

Unit 2 Title: WSN Architecture (7 Hrs)

U2.1 WSN Architectural Framework: Sensor network Senarios: types of sources and sinks, single-hop versus multihop networks, types of mobility (node, sink & event) and optimization goals, Physical layer transceiver design consideration in WSNs. Fundamentals of wireless MAC protocols, Low duty cycle protocols and wakeup concept, contention-based protocols, Schedule-based protocols. Network layer – routing metrics, flooding and gossiping, routing protocols. Transport Layer - Data-Centric and Contention-Based Networking, QoS in Wireless Sensor Networks – Congestion Control, QoS management: Basic functions, centralized solution, Topology control, Sensor mode selection.

U2.2 Self Study: Physical layer – Wireless channel and communication fundamentals: Channel models, wave propagation effects and noise, IEEE 802.15.4 MAC protocol,

Unit 3 Title: Time Synchronization and Localization in WSN (7 Hrs)

U3.1 Node and Network Management: power management – local power management, dynamic power management, clock synchronization in WSN – challenges, design principle, classification and protocols (Lightweight Tree-Based Synchronization, Time-diffusion based synchronization, Asynchronous diffusion based). Localization – classification of localization,

localization algorithms: ranging techniques, range-based localization, range-free localization, event-driven localization.

U3.2 Self Study: Application of Clock synchronization and localization algorithms in WSN, Beacon based distributed localization

Unit 4 Title – Wireless Security (7 Hrs)

U4.1 WSN Security: Security fundamentals and challenges, Static and dynamic key distribution, security attacks, protocols and mechanisms for security. Different types of attacks in WSN, WPA-EAP, Attacking 802.11 Networks- Basic Types Of Attacks, Security Through Obscurity, Defeating WEP, WEP attacks, 802.11 Authentication Types, Attacking WPA Protected 802.11, Breaking WPA, LEAP, EAP-TLS, Tunnelling EAP Techniques, Hacking Attacking 802.11i wireless technologies- Hacking hotspots, client attacks resources, threats of Bluetooth.

U4.2 Self Study: Advanced attacks- layer 2 fragmentations breaking the silence, layer 2 and layer 3 resolutions.

Unit 5 Title – Programming in WSN (7 Hrs)

U5.1 Sensor network programming: Radio basics, introduction to ZigBee – network topology, addressing basics, PAN addresses, channels, basic ZigBee chat, advanced ZigBee, introduction to Arduino, Embedded system programming concept and I/O interfacing using Arduino. Inter building WSN with Zigbee and Arduino. Important roles of Arduino based systems in smart cities.

U5.2 Self Study: Practical implementation of WSN on environment monitoring, Mobile Robot motion control using Arduino based system.

Text Books:

1. Holger Karl & Andreas Willig, " Protocols And Architectures for Wireless Sensor Networks" , John Wiley, 2005.
2. Feng Zhao & Leonidas J. Guibas, “Wireless Sensor Networks- An Information Processing Approach", Elsevier, 2007.

Reference Books

1. Ian F. Akyildiz, Mehmet Can Vuran, *Wireless Sensor Networks in Communications and Networking*, Wiley, 2011.
2. Robert Faludi , *Building Wireless Sensor Networks: with ZigBee, XBee, Arduino, and Processing*, O'Reilly Media, 2010.
3. Ibrahiem M. M. El Emary, S. Ramakrishnan, *Wireless Sensor Networks: From Theory to Applications*, CRC Press, 2013.
4. Waltenegus Dargie, Christian Poellabauer, *Fundamentals of Wireless Sensor Networks: Theory and Practice*, Wiley-Blackwell, 2010.

Course Outcomes:
Upon completion of the course the student will be able to-

CO1 : Identify the unique issues in sensor networks.

CO2: Implement and deploy current technology trends for wireless sensor networks.
CO3: Analyze the challenges in designing MAC, routing and transport protocols for wireless sensor networks.
CO4: Understand and apply various security issues pertaining to wireless sensor networks.
CO5: Comprehend and exploit the various sensor network Platforms, tools and applications.

COURSE CODE: IT42323 **REF NO:** To be filled by CD office

WIRELESS SENSOR NETWORK LABORATORY

Credits: 01 **Teaching Scheme:** Laboratory 02Hrs/Week

Prerequisites:

Data Communication and Computer Networks
Computer Programming
Design & Analysis of Algorithms
Operating System

Objectives:

1. To understand the fundamental principles of wireless sensor networks.
2. To create different topologies in wireless sensor network
3. To simulate different algorithms by using simulation software

List of Experiments:

Experiment No. 1

- Introduction to Sensor node: To study the various components of a sensor node and their functionalities.

Experiment No. 2 (NS2/NS3)

- Introduction to Simulator: Study on various features of the simulator to design a sensor network.

Experiment No. 3 (NS2/NS3)

- Write the simple programs to create an arbitrary topology sensor network.

Experiment No. 4 (NS2/NS3)

- Write a program to send data (Hello Message) among the sensor nodes using unicast, broadcast and multicast techniques.

Experiment No. 5

- Write a program to implement the Ad-Hoc on Demand Distance Vector routing algorithm.

Experiment No. 6

- Write a program to implement various propagation loss models on the sensor network.

Experiment No. 7

- Write a program to calculate the energy of the sensor nodes after transmitting n number of messages.

Experiment No. 8

- Write a program to measure the various network performance parameters (such as delay, clock time of the sensor node, network life time, throughput etc.) of the sensor nodes after transmitting N number of messages.

Experiment No. 9

- Design an arbitrary topology sensor network of N senor nodes using C Mote or IR sensor nodes.

Experiment No. 10

- Write a program to collect the environment data (such as temperature, pressure, humidity etc.) from the sensor node and send it to the base station for further processing.

Text Books:

T1. Holger Karl & Andreas Willig, " Protocols And Architectures for Wireless Sensor Networks" , John Wiley, 2005.
T2. https://www.nsnam.org /

Course Outcome:
After taking this course the graduate students will be able to:
1. Identify different features supported by the sensor networks.
2. Design and implementation of sensor network using simulation software.
3. Analyze and implement data communication and routing using socket programming concepts
4. Design and implementation of sensor network using sensor nodes

COURSE CODE: CS42175 **REF NO:**

Mobile Computing

Credits:3 **Teaching Scheme: -** Theory 3 Hrs/Week

Prerequisites:

1. Java Programming
2. HTML and XML scripting language
3. Basic idea about mobile technology and networking

Objectives:

1. To develop the skills required to create applications for mobile phone
2. To learn how to effectively use application architectures to build a variety of Mobile Apps.
3. To develop skills for processes for mobile software engineering

Course Details:

Unit 1 **(8 Hrs)**

Title- Fundamentals of Mobile Technology

U1.1

Basic wireless technology, overview of cellular communications, components of cellular system (BTS,BSC,MSC), protocol stack and information flow (call processing, MOC and MTC), SMS concept, Network Architeture, message processing, Overview of CDMA (Concept, network components, architecture), Digital techniques using spread spectrum, advantages and disadvantages. Overview of 2.5G and 3.0G, Need,concepts, architecture and applications (GPRS,UMTS, 3GPP, WCDMA, HSCSD), Introduction to GSM, standards, services offered,architecture, functionalities of components, mobile registration process, Handover,GSM channels, traffic, control channel, Radio frequency power levels, Timimg in advance and power control, burst structure (normal and random access),

U1.2

Self Study: frequency correction and synchronization, frame structures (traffic channel, signaling frame), different call scenarios along with traffic and control channels.

Unit II

Title- Mobile Communication and Management:

U2.1 **(8 Hrs)**

Mobile originated voice call, terminated voice call, location update procedure, sending and receiving SMS, speech and channel coding, control channel / logic channels, BCCH,CCCH,RACH,SDCCH,AGCH,PCH,SAACH,FACCH, 2.5G /3G system, evolution of GPRS, Its network elements, applications, architecture, QOS, mobility management, comparison with HSCSD, spread spectrum for 3G, network elements, data speeds,

applications,WLAN (standard a/b/g), security, applications, IAPP, mobile IP, VOIP (Functionality, shortcomings, legal issues), NFC.

U2.2

Self Study: Introduction to QoS, integration with GPRS, Bluetooth and 3G

Unit III **(7 Hrs)**

Title- Android, Its Applications and Services

U3.1

Introduction to android, why android, activities and tasks, android SDK features,development environment of SDK, creating android activities, fundamental android UI design, Intent, adapter, dialog, android technique for saving data, data base in android, data storage and contents providers, maps, Geocoding, location based services, Toast, alarms, instant messaging using Bluetooth, telephony using Bluetooth, introduction to sensor manager, managing network and wi-fi connection, advanced android development.

U3.2

Self Study: UI based design (chat, and mail application).

Unit IV **(7 Hrs)**

Title- Application Development using Android

U4.1

Linux kernel security, implement AIDL interface, play audio and video, playing from raw resource, file or stream, audio capture, simple graphics inside a view, Gaming development-: introduction to OpenGL, scalable 2D / 3D graphics API, drawables, shape drawable, web services API(JSR-172), creating from resource images, resource XML, Nine-patch,tween animation, frame animation

U4.2

Self Study: web services API its applications and usability

Unit V **(6 Hrs)**

Title- Activity,Process and service

U5.1

Activity lifecycle, configuration changes, starting activity and getting results, saving persistent state, permissions,process lifecycle,process to set default actions for activities,passing information between activities, content provider basics,modifying data in a provider, creating a content provider, content URI summary, receiver lifecycle,permissions, introduction to service, service lifecycle, permissions, local service sample, remote messenger service sample.

U5.2

Self Study: Augmented reality of application, integration with social networking website

Text Books

T1."Android Application Developmet"Rick Roger, O'reilly, 2009

T2. "Mobile and wireless Design Essential", Martyn Mallick, John, Wiley & Sons 3rd Edition.

Reference Books

R1. Introduction to Android Application Development Joseph Annuzzi, Jr. Lauren Darcey

Shane Conder, Addison-Wesley 4th Edition.

R2. Fundamentals of mobile computing Prasant Kumar Pattnaik and Rajib Mall PHI.

R3. Wireless Communications & Networks William Stallings Pearson

Course Outcomes:

Upon completion of the course, graduates will be able –

CO1: Interpreting the fundamental technology of mobile technology and its enhancement

CO2:Differentiating the mobile communication and its management by techniques and methods

CO3: Exemplifying and Summarizing the functionality and services of android for mobile application development.

CO4: Implementing and Executing advanced application developments.

CO5: Attributing and Organizing activity, process and services of reality of applications

COURSE CODE: CS42375 **REF NO: To be filled by CD office**

Mobile Computing Lab (Mobile Application Development)

Credits: 01 **Teaching Scheme: -** Laboratory 02 Hrs/Week

Prerequisites:

4. Java Programming
5. HTML and XML scripting language
6. Basic idea about mobile technology and networking

Objectives:

4. To understand features of Android operating system for mobile phone sets.
5. To develop the skills required to create applications for mobile phone.
6. To develop mobile applications using Java-Android programming for various problem areas

List of Practicals:

Topic 1: Introduction to Android

Experiment 1: Write a Hello world program on android.

Experiment 2: Implement a lifecycle of an android application using toast.

Experiment 3: Implement a simple calculator using button action listener.

Topic 2: Android Activities& UI design

Experiment 4: Create an application which implements data transfer between activities.

Experiment 5: Create an application which implements multiple spinners dependent on each other.

Experiment 6: Create an application which shows a list of name and images together.

Experiment 7: Create an application which shows images in grid view. When you long press on image, it will show you options remove and zoom. When you touch on zoom it will zoom the image on another activity. When you touch on remove it will remove from grid view.

Topic 3: Android Techniques for saving data

Experiment 8: Create an application having user registration & login using SharedPreference.

Experiment 9: Create an application which show you a list of name and age together when you click on the add option menu, another dialog box open. Fill the details and when you press ok it will load in the list view.

Experiment 10: Develop an application that store SMS from particular number in your own database.

Topic 4: Services, Maps, Geo-coding & Location Based Services in Android

Experiment 11: Create a service that when called stores missed call numbers in database.

Experiment 12: Create a service that executes when there is change in location & store the location in database with following fields: id, latitude, longitude & time.

Experiment 13: Create an application in which a service executes every time when the system boot completes and send user GPS location to specific mobile number.

Note: Any 10 experiments to be done from the above list

Course Outcomes:

Upon completion of the course, graduates will be able to:

1. Interpret the fundamental technology of android programming
2. Implement and execute user interface based applications.
3. Analyze and develop the functionality and location based services of android for mobile application development.

COURSE CODE: IT42175 **REF NO: To be filled by CD office**

Bioinformatics

Credits: 3 **Teaching Scheme: -** Theory 3 Hrs/Week

Prerequisites:

1. Design and analysis of algorithm
2. Data structures
3. Machine Learning
4. Computer programming

Objectives:

1. To know the recent evolution in biological science.
2. To make students aware about computational problems lie on biology.
3. To improve the programming skills in bioinformatics domain.

Course Details:

Unit 1
Title- Introduction to Bioinformatics:

(06 Hrs)

U1.1
Introduction to Bioinformatics and Computational Biology, Biological sequences, Biological databases, Genome specific databases, Data file formats, Data life cycle, Sequence Analysis, Pair-wise alignment.

U1.2. Self Study: Case study on a genomic database.

Unit 2
Title – Sequence alignment algorithms **(08Hrs)**

U2.1
Dynamic programming algorithms for computing edit distance, string similarity, shotgun DNA sequencing, end space free alignment. Multiple sequence alignment, Algorithms for Multiple sequence alignment, Generating motifs and profiles, Local and Global alignment, Needleman and Wunsch algorithm, Smith Waterman algorithm, BLAST and PSIBLAST algorithms.

U2.2. Self Study: Study of PHIBLAST tool and its algorithm.

Unit 3
Title – Phylogenetics and Protein structure prediction **(08 Hrs)**
U3.1
Introduction to phylogenetics, Distance based trees UPGMA trees, Molecular clock theory, Ultrametric trees, Parsimonious trees, Neighbor joining trees, trees based on morphological traits, Bootstrapping. Protein Secondary structure and tertiary structure prediction methods, Homology modeling, abinitio approaches, Threading, Critical Assessment of Structure Prediction.

U3.2. Self Study: Case study on Structural genomics.

Unit 4
Title – Applications of machine learning techniques in bioinformatics problem
(08 Hrs)

U4.1
Machine learning techniques: Artificial Neural Networks in protein secondary structure prediction, Hidden Markov Models for gene finding, Decision trees, Support Vector Machines.

U4.2. Self Study: Study of using artificial neural network for tertiary protein structure prediction

Unit 5
Title- Systems Biology and Synthetic Biology
(08 Hrs)

U5.1
Introduction to Systems Biology and Synthetic Biology, Microarray data analysis and bi-clustering, DNA computing, Bioinformatics approaches for drug discovery, Applications of informatics techniques in genomics and proteomics: Assembling the genome, STS content mapping for clone contigs, Functional annotation, Peptide mass fingerprinting.

U5.2. Self Study: Exploring other applications of informatics genomics and proteomics.

Text Books:

1. Lesk, A. K., "Introduction to Bioinformatics" 4th Edition, Oxford University Press, 2013
2. Dan Gusfield, "Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology" Cambridge University Press, 1997.
3. Durbin, R., Eddy, S., Krogh, A., and Mitchison, G., "Biological Sequence Analysis Probabilistic Models of proteins and nucleic acids" Cambridge, UK: Cambridge University Press, 1998.
4. Mount, D.W., "Bioinformatics Sequence and Genome Analysis" 2nd Edition, Cold Spring Harbor Laboratory Press, 2004

Reference Books:

1. Baldi, P. and Brunak, S., "Bioinformatics: The Machine Learning Approach" 2nd Edition, MIT Press, 2001.
2. Tindall, J., "Beginning Perl for Bioinformatics: An introduction to Perl for Biologists" 1st Edition, O'Reilly Media, 2001

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Identify all underlying computational problems in Biology.
2. Use and apply bioinformatics tools for solving computational issues.
3. Apply computational based solutions for biological perspectives.
4. Pursue higher education in this field.
5. Practice life-long learning of applied biological science.

COURSE CODE: IT42175 **REF NO: To be filled by CD office**

Bioinformatics Tutorial

Credits: 01 **Teaching Scheme: -** Tutorial 01 Hrs/Week

Prerequisites:
1. Computer programming
2. Data structures
3. Design & analysis of algorithm
4. Machine Learning

Objectives:

1. To know the recent evolution in biological science.
2. To make students aware about computational problems lie on biology.
3. To improve the programming skills in bioinformatics domain.

Course Details:

List of Contents:

Problems solving and exercises on the following topics:

Tutorial No. 1: Basics of Bioinformatics.

Tutorial No. 2: Understanding of data files formats

Tutorial No. 3: Problem solving on Pair wise sequence alignment algorithms.

Tutorial No. 4: Problem solving on multiple sequences alignment algorithms.

Tutorial No. 5: Construction of phylogenetics tree.

Tutorial No. 6: Problem solving on Protein structure prediction.

Tutorial No. 7: Application of ANN in prediction of protein structure.

Tutorial No. 8: Understanding Hidden Markov Models for gene finding.

Tutorial No. 9: Understanding biology and DNA computing.

Tutorial No. 10: Applications of informatics techniques in genomics and proteomics.

Text Books:

1. Lesk, A. K., "Introduction to Bioinformatics" 4th Edition, Oxford University Press, 2013
2. Dan Gusfield, "Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology" Cambridge University Press, 1997.
3. Durbin, R., Eddy, S., Krogh, A., and Mitchison, G., "Biological Sequence Analysis Probabilistic Models of proteins and nucleic acids" Cambridge, UK: Cambridge University Press, 1998.

4. Mount, D.W., "Bioinformatics Sequence and Genome Analysis" 2nd Edition, Cold Spring Harbor Laboratory Press, 2004

Reference Books:

1. Baldi, P. and Brunak, S., "Bioinformatics: The Machine Learning Approach" 2nd Edition, MIT Press, 2001.
2. Tindall, J., "Beginning Perl for Bioinformatics: An introduction to Perl for Biologists" 1st Edition, O'Reilly Media, 2001

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Identify all underlying computational problems in Biology.
2. Use and apply bioinformatics tools for solving computational issues.
3. Apply computational based solutions for biological perspectives.
4. Pursue higher education in this field.
6. Practice life-long learning of applied biological science.

COURSE CODE: IT42375 **REF NO: To be filled by CD office**

Python Programming Lab
(Against Bioinformatics Lab Elective)

Credits: 01 **Teaching Scheme: -** Laboratory 02 Hrs/Week

Prerequisites:

1. Computer programming.
2. Object oriented programming knowledge

Objectives:

1. Understanding the Fundamentals of Python Programming.
2. Understanding object oriented concepts in Python
3. To make familiar with Python Functions and Modules.
4. To access and manipulate Python Files
5. To access and retrieve data from various data bases through Python.

List of Practicals:

Experiment No. 1: Getting familiar with Variables, expressions and statement
a. Python programming environment
b. Python Environment setup
c. Python basic syntax
d. Variables
c. Operators

Experiment No. 2: Program to understand Python decision making and Loops.

Experiment No. 3: Programs to understand Python Strings

Experiment No. 4: Program to understand Python Data structures.
Write and execute program using data structures like
(a) List
(b) Tuple
(c) Dictionary

Experiment No. 5: Programs to implement Python Functions.
(a) Composing expressions
(b) Function definitions and use
(c) Flow of execution
(d) Parameters and arguments.

Experiment No. 6: Program to use Python Exceptions

Experiment No. 7: Program to implement Python classes and Objects.
(a) Object class
(b) Python constructor
(c) Python Inheritance

Experiment No. 8: Program to implement Python Files

(a) Files
(b) Processing things from a file.
(c) Directories
(d) Modules
(e) Creating Modules

Experiment No. 9 & 10: Python Programs to do database access.
(a) Python Database API to connect to and access Oracle database
(b) Python Database API to connect to and access MySql database

Text Books:
1.Python (3.6) Tutorial (Guiddo Van Rossume, 2016).

Reference Books:

Course Outcomes:

Upon completion of the course, graduates will be able to-

1. Apply fundamental python programming concepts to do simple programming.
2. Use the object oriented programming concepts to solve simple programming assignment
3. Use Python data structures to manipulate different types of data.
4. Access files and manipulate files in Python
5. Retrieve and manipulate data from Oracle and MySQL Database using Python.

COURSE CODE: CS42172 **REF NO: To be filled by CD office**

DIGITAL IMAGE PROCESSING

Credits: 03 **Teaching Scheme:** Theory 03 Hrs/Week

Prerequisites: Knowledge on engineering mathematics, programming and algorithms.
Knowledge on Digital signal processing is preferable.

Objective:

- To gain knowledge on the different techniques used for digital image processing
- To apply DIP techniques for solution of different type of problem domain.

Course Details:

Unit I: Digital Image fundamentals **(6Hrs)**
U1.1 Introduction: Image sampling and quantization, relationship between pixels, Intensity transformations and spatial filtering, some basic intensity transformation functions, Histogram processing, spatial filters for smoothing and sharpening

U1.2. Self Study:
Visual Perception, Image acquisition, using fuzzy sets for intensity transformation and spatial filtering

Unit 2: Frequency Domain Filtering and Image Restoration (8Hrs)
U2.1 Filtering in the Frequency Domain: preliminary concepts, 2D DFT and its properties, FFT, basic filtering in the frequency domain, image Smoothing- Ideal LPF, Butterworth LPF, image Sharpening- Ideal HPF, Butterworth HPF

U2.2 Image Restoration and Reconstruction: Noise models: Spatial and frequency properties of noise, Noise PDF, estimation of Noise parameters,, restoration in the presence of noise only-Mean and Order Statistics filters, estimating the degradation function

U2.3. Self Study
Bandpass and Bandreject Filters, Separability of 2-D DFT and FFT, Adaptive restoration, Inverse Filtering

Unit 3: Image Segmentation (6 Hrs)
U3.1: Point, Line and Edge Detection: Isolated points, Line Detection, Edge models, Edge linking and boundary detection, Global thresholding, Multiple and variable thresholding

U3.2: Region based segmentation: Region growing, Region splitting and merging

U3.3 Self Study
Watershed Segmentation algorithm, Use of motion in segmentation (spatial and frequency techniques)

Unit 4: Image Compression (8 Hrs)
U4.1: Image Compression using block transforms: Fundamentals, Some basic compression methods – Huffman Coding, LZW coding, Run-Length coding, Predictive coding, Block transform coding

U4.2 Wavelets and Multi-resolution Processing: multi-resolution expansions, wavelet transforms in one and two dimension, Compression using Wavelet transforms

U4.3 Self Study:
Arithmetic Coding, Bit –Plane coding, Delta Modulation and DPCM , Fast Wavelet Transform, JPEG standards

Unit5: Color processing and Morphological Image processing (6Hrs)

U5.1 Color Image Processing: Color models (RGB, HSI, CMY etc.), Model Conversions, Color transformation (Histogram Processing etc.)

U5.2: Morphological Image Processing: Erosion and Dilation, Basic morphological algorithms-boundary extraction, hole filling, connected components, Opening and Closing

U5.3 Self Study:
Color image smoothing and sharpening, Morphological Thinning, Thickening, Pruning,

Text Books:
1. "*Digital Image Processing"*, R.C. Gonzalez, R.E. Woods, 3rd Edition, Pearson Education
2. "*Digital Image Processing using Matlab"*, R C Gonzalez, Woods and Eddins, 2nd Edition, Tata McGraw Hill
References:
1. "*Digital Image Processing"*, S.Sridhar, Oxford University Press, 2011

Course Outcomes:

Upon completion of the course the graduate students will be able to

1. Use intensity transformation functions and perform spatial filtering for smoothing and sharpening of digital images
2. Design low pass, high pass and band pass frequency domain filtering. Apply noise modeling and restoration techniques for reconstruction of original images from noisy image.
3. Understand color image models. Use the wavelet concept to design image processing filters.
4. Use coding techniques for image compression. Apply morphological operations for image understanding, reconstruction and object detection.
5. Analyze and perform suitable segmentation method for detection image objects.

COURSE CODE: CS42172 **REF NO: To be filled by CD office**

DIGITAL IMAGE PROCESSING – TUTORIAL

Credits: 01 **Teaching Scheme:** Theory 01 Hrs/Week

Prerequisites: Knowledge on engineering mathematics, programming and algorithms.
Knowledge on Digital signal processing is preferable.

Objective:

- To gain knowledge on the different techniques used for digital image processing
- To apply DIP techniques for solution of different type of problem domain.

Course Details:

Problem solving and exercises on the following topics:

Tutorial 1: Image sampling and quantization, pixel relationship
Tutorial 2: Image operations, Distance measures
Tutorial 3: Intensity transformations and Histogram processing
Tutorial 4: DFT computation in 1-D and 2-D
Tutorial 5: Filtering in spatial and frequency domain
Tutorial 6: Image Restoration - Order statistics filtering
Tutorial 7: Colour models, Colour transformation
Tutorial 8: Image coding – RLC, Huffman, LZW, Block transform and coding
Tutorial 9: Line and Edge, Boundary detection, Segmentation techniques
Tutorial 10: Morphological operations
Tutorial 11: Wavelet transforms in one and two dimension

Text Books:
1. "*Digital Image Processing"*, R.C. Gonzalez, R.E. Woods, 3rd Edition, Pearson Education
2. "*Digital Image Processing using Matlab"*, R C Gonzalez, Woods and Eddins, 2nd Edition, Tata McGraw Hill

References:
1. "*Digital Image Processing"*, S.Sridhar, Oxford University Press, 2011

Course Outcomes:

Upon completion of the course the graduate students will be able to

1. Use intensity transformation functions and perform spatial filtering for smoothing and sharpening of digital images
2. Design low pass, high pass and band pass frequency domain filtering. Apply noise modeling and restoration techniques for reconstruction of original images from noisy image.
3. Understand color image models. Use the wavelet concept to design image processing filters.
4. Use coding techniques for image compression. Apply morphological operations for image understanding, reconstruction and object detection.
5. Analyze and perform suitable segmentation method for detection image objects.

COURSE CODE: CS42372 **REF NO: To be filled by CD office**

Digital image processing Lab

Credits: 01 **Teaching Scheme: -** Laboratory 02 Hrs/Week

Prerequisites:
1. Computer programming.

Objectives:

1. Understanding the Fundamentals of computational tools such as Mat Lab/ Scilab.
2. Application of Pixel operations, Arithmetic and Logical operations on image.
3. Application of histogram processing, image enhancement and restoration.
4. To implement filtering operations on images.
5. Use image processing techniques for image segmentation.

List of Practicals:

Experiment No. 1: Fundamental commands on image using Mat Lab
- a. How to read and show an image in Mat lab.
- b. How to access Image Pixels in Mat lab.
- c. How to write Image in Mat lab.

Experiment No. 2: Program to understand Arithmetic and logical operations on Image.
- a. Write Program to read any image, resize it to 256_256. Apply square mask so that only middle part of the image is visible.
- b. Write an user defined matlab function addbrightness() and use it to increase brightness of given image. Write a program to demonstrate watermarking using EX-OR operation.
- c. Prepare any two images of size 256 _ 256 in paint. Save it in JPEG f format 256 gray levels. Perform logical NOR, NAND operations between two images. Write program and paste your results.

Experiment No. 3: Programs to understand image enhancement.
- a. Write a program to compute the histogram of an input image
- b. Program for calculation and equalization of the histogram
- c. Take a photograph in dark area. Improve its appearance using histogram equalization technique.
- d. Use mean, median, Max and Min filters for image smoothing

Experiment No. 4: Program to perform Geometric transformation on image.
Write and execute program for geometric transformation of image
- (a) Translation
- (b) Scaling
- (c) Rotation
- (d) Shrinking
- (e) Zooming

Experiment No. 5: Programs to implement Image Restoration.
- a. To add noise in the image and apply image restoration technique using median filter.

b. Write a program for removing salt and pepper noise for restoring the image.

Experiment No. 6: Program to understand filtering of images

a Use 3x3 Mask for low pass filter and high pass filter.

b. Write a program to implement spatial filtering.

c. Perform Low pass and high pass filtering in frequency domain.

Experiment No. 7: Program to implement morphological operations.

a. Write a program to demonstrate Erosion and Dilation.

Experiment No. 8: Program to implement image segmentation.

a. Write a program to image segmentation based on region growing approach.

b. Write a program for object detection using threshold technique.

Experiment No. 9: Programs for edge detection

a. Write a program for various edge detection techniques.

b. Write a program to compare and identify which edge detection technique is better.

Text Books:

1. Digital Image Processing using Mat Lab, Rafel C Gonzalez, TMH, 2nd Edition.

Reference Books:

Course Outcomes:

Upon completion of the course, graduates will be able to-

6. Perform different operations to manipulate pixels on images.
7. Design and implement Arithmetic and Logical operations on image.
8. Analyze the images in original and enhanced mode.
9. Implement spatial domain and frequency domain operations.
10. Analyze and design image restoration and image segmentation technique.

COURSE CODE: IT42115 **REF NO: _____**

Real Time Systems

Credits: 3 **Teaching Scheme: -** Theory 3 Hrs/Week

Prerequisites:

1. Basic Understanding of Operating Systems
2. Basic Understanding of Computer Architectures
3. Basic Networking Knowledge
4. Knowledge of Programming Language

Objectives:

1. To expose the students to the fundamentals of real time systems
2. To analyze the various scheduling aspects of real time systems
3. To study about the resource access control and synchronization in real time systems
4. To introduce the fundamentals of real time communication
5. To study the data management system for real time

Course Details:

Unit 1

Title - Introduction: (08 Hrs)

U1.1.

Introduction to Real Time Systems, Characteristics of Real Time System, Applications of Real Time Systems, A Basic Model of a Real Time System, Types of Real Time Systems, Fault Tolerance Techniques in Real Time Systems: Hardware Fault Tolerance and Software Fault Tolerance, Timing Constraints: Classification and Examples of Different Types of Timing Constraints.

U1.2. Self Study: Modelling of Timing Constraints

Unit 2

Title - Real Time Task Scheduling: (8 Hrs)

U2.1.

Basic Concepts and Terminologies, Classification of Real Time Task Scheduling Algorithms, Clock Driven Scheduling, Event Driven Scheduling, Earliest Deadline First (EDF) Scheduling, Rate Monotonic Algorithm (RMA) Scheduling, Issues Associated with RMA.
Commercial Real Time Operating Systems: General Concepts, Unix and Windows as RTOS.

U2.2. Self Study: Hybrid schedulers, Survey on Commercial Real Time Operating Systems

Unit 3

Title - Resource Sharing and Multiprocessor Scheduling in RTOS: (8 Hrs)

U3.1.

Resource Sharing Among Real Time Tasks, Priority Inversion, Priority Inheritance Protocol (PIP), Highest Locker Protocol (HLP), Priority Ceiling Protocol (PCP),Types of Priority Inversions Under PCP, Important Features of PCP.

Scheduling Real Time Tasks in Multiprocessor and Distributed Systems: Multiprocessor Task Allocation, Dynamic Allocation of Tasks, Clock Synchronization.

U3.2. Self Study: Handling Task Dependencies, Issues in Using a Resource Sharing Protocol

Unit 4
Title - Real Time Communication: (06 Hrs)

U4.1.
Basic Concepts, Real Time Communication in a LAN: Hard Real Time Communication and Soft Real Time Communication, Real Time Communication over Packet Switched Networks.

U4.2. Self Study: QoS Framework

Unit 5
Title - Real Time Databases: (06 Hrs)

U5.1.
Review of Basic Database Concepts, Real Time Databases, Concurrency Control in Real-Time Databases.

U5.2. Self Study: Characteristics of Temporal Data, Example Applications of Real-Time Databases

Note: Five assignments to be given to the students on self study, comprising of one assignment from each unit.

Text Books:
1. Rajib Mall, "Real Time Systems Theory and Practice", Pearson Publication, 2008
2. Jane W.S. Liu, "Real Time Systems", Pearson Education, 2000
3. C.M. Krishna and K.G. Shin, "Real-Time Systems", TMH

Reference Books:
1. R.J.A Buhur, D.L Bailey, "An Introduction to Real – Time Systems", Prentice – Hall International, 1999
2. Phillip A. Laplante, "Real-Time Systems Design and Analysis", Wiley Publishers

Course Outcomes:

1. Understand the basics and importance of real time systems
2. Develop real time algorithm for task scheduling
3. Apply formal methods to the analysis and design of real time systems
4. Explain the additional problems that arise in developing distributed and networked real-time systems
5. To work on design and development of protocols related to real time communication
6. To know the details of real time databases and applications

COURSE CODE: IT42115 **REF NO: _____**

Real Time System – Tutorial

Credits: 1 **Teaching Scheme: -** Tutorial 1 Hrs/Week

Prerequisites:

1. Operating System,
2. Theory of computation,
3. Computer architecture

Objectives:

1. Understand the fundamental concept of real time systems
2. Study the design principles of real time system
3. Understand the different scheduling mechanisms
4. Design and implementation of different task scheduling policies
5. Understand real time operating systems and its principles.

Course Contents:

Excercises / Problem solving will be carried out based on the following topics:

Tutorial No. 1: Development of different Timing Constraint Model.

Tutorial No. 2: Concept of Clock-Driven Scheduling with numerical on Table-Driven Scheduling and Cyclic Scheduler.

Tutorial No. 3: Understanding the concept of Event-Driven Schedulers using problematic approach: Earliest Deadline First (EDF) and Rate Monotonic Algorithm (RMA).

Tutorial No. 4: Understanding the need of DMA with/without using Context Switching Overhead and Self Suspension.

Tutorial No. 5: Concept of resource sharing among real-time system using Priority Inversion, Priority Inheritance Protocol and Highest Locker Protocol.

Tutorial No. 6: Understanding the concept of Priority Ceiling Protocol and Handling Task Dependencies.

Tutorial No. 7: Developing the concept of Multiprocessor and Dynamic Allocation of Task in real time system.

Tutorial No. 8: Working of Centralized and Distributed clock synchronization techniques.

Tutorial No. 9: A Survey of Contemporary Real-Time Operating Systems.

Tutorial No. 10: Understanding the concept of Centralized and Distributed clock synchronization techniques.

Tutorial No. 11: Understanding soft and hard real time communication in a LAN.

Tutorial No. 12: Understanding the application of real-time database using a commercial real-time database.

Text Books: As in RTS theory

Reference Books: As in RTS theory

Course Outcomes: As in RTS theory

COURSE CODE: CS40111 **REF NO: To be filled by CD office**

Embedded System

Credits: 3 **Teaching Scheme: -** Theory 3 Hrs/Week

Prerequisites:

1. Computer Organization
3. Operating Systems
4. Computer Programming

Objectives:

1. To expose the students to the fundamentals of embedded systems
2. To demonstrate in-depth knowledge in hardware and software used in embedded systems
3. To analyze the various scheduling aspects of real time systems
4. To understand the design procedure of embedded firmware
5. To know about the embedded software development procedure and testing methods.

Course Details:

Unit 1

Title – Introduction to Embedded System: (06 Hrs)

U1.1.

Definition of Embedded System, Embedded Systems Vs General Computing Systems, History of Embedded Systems, Classification, Major Application Areas, Purpose of Embedded Systems, Basic Characteristics of an Embedded System.

U1.2. Self Study: Operational and non-operational quality attributes

Unit 2

Title – Overview of Processors and Hardware Concepts: (9 Hrs)

U2.1.

General purpose processors and Domain Specific Processors, Microcontrollers, ARM-based Systems on a Chip (SoC), Application-Specific Circuits (ASICs), VHDL, Sensors, A/D-D/A converters, Actuators, Interfacing using UART, USB, CAN bus, SRAM and DRAM, Flash memory.

U2.2. Self Study: Microprocessor Vs Microcontroller, Commercially available Microprocessor and Microcontroller, Levels of hardware modelling

Unit 3

Title - Real Time Task Scheduling: (9 Hrs)

U3.1.

Basic Concepts and Terminologies, Classification of Real Time Task Scheduling Algorithms, Clock Driven Scheduling, Event Driven Scheduling, Earliest Deadline First (EDF) Scheduling, Rate Monotonic Algorithm (RMA) Scheduling, Issues Associated with RMA, Hybrid schedulers.

U3.2. Self-Study: Case studies on Commercial Real Time Operating Systems: General Concepts, Unix and Windows as RTOS.

Unit 4

Title – Embedded System Development and Design: (06 Hrs)

U4.1.

Embedded system development life cycle, General language characteristics , Features of MISRA C for embedded programming, Hardware/Software Co-design, Hardware/software partitioning.

U4.2. Self Study: State charts, Trends in embedded industry

Unit 5

Title – Testing Embedded Systems: (06 Hrs)

U5.1.

Software Tools, Host and Target Machines, Linkers/Locators for Embedded Software, Getting Software into the Target System, Embedded systems Testing, Design for testability and Self-test.

U5.2. Self Study: Simulation and Debugging Tools and Techniques

Text Books:

T1. Rajkamal, "Embedded Systems Architecture, Programming and Design", TATA McGraw-Hill

T2. Frank Vahid and Tony Givargis," Embedded Systems Design – A unified Hardware /Software Introduction", John Wiley, 2002

T3. Rajib Mall, "Real-Time Systems Theory and Practice", Pearson Education

Reference Books:

R1. Shibu KV, "Introduction to Embedded Systems", TMH

R2. S. Chattopadhyay, "Embedded System Design", PHI

R3. David E.Simon," An Embedded Software Primer", Pearson Education Asia, First Indian Reprint 2000

Course Outcomes:

Upon completion of the course the graduates will be able-

CO1. To understand the basics and importance of embedded systems

CO2. To understand the selection procedure of processors in the embedded domain

CO3. To visualize the role of real time operating systems in embedded systems

CO4. To contributepositively in developing solutions with embedded systems for multidisciplinary scientific problems with open mindedness, objectivity and rational approach

CO5. To analyze the various techniques and tools used for testing and debugging

COURSE CODE: CS42173 **REF NO: _____**

Distributed Operating System

Credits: 3 **Teaching Scheme: -** Theory 3 Hrs/Week

Prerequisites:

1. Operating System.

Objectives:

- To gain knowledge on Distributed operating system concepts that includes architecture, Mutual exclusion algorithms, Deadlock detection algorithms and agreement protocols
- To gain insight on to the distributed resource management components. The algorithms for implementation of distributed shared memory, recovery and commit protocols

Course Details:

Unit 1

Title- Process Synchronization: (06 Hrs)

U1.1.

Concept of processes, Concurrent processes, Threads, Overview of different classical synchronization problems, Monitors, Communicating sequential processes (CSP)

PROCESS DEADLOCKS: Introduction causes of deadlocks, Deadlock handling strategies Models of deadlock. Distributed deadlock algorithms for Avoidance, Prevention, and Detection.

RPC: RPC Model, Transparencies in RPC, Implementation, Stub Generation, RPC Messages, Server Management, Call Semantics, Communication Protocols

U1.2. Self Study: Classification of distributed deadlock detection algorithms, Distributed Objects: Remote Method Invocation, Case studies of existing RPC implementation.

Unit 2

Title – Basics of Distributed Operating System: (08 Hrs)

U2.1.

Architectures, Issues in Distributed operating systems, Limitations of Distributed Systems, Temporal ordering of events, Lamport's logical clock, Global states, Chandy-Lamport's global state recording algorithm.

Co-ordination: Election Algorithm: Bully Algorithm, Ring Algorithm

Basic concepts of Distributed Mutual Exclusion, Lamport's Algorithm, Ricat-Agrawala Algorithm: Basic concepts of Distributed deadlock detection

U2.2. Self Study: Network operating system: Design issues, working principles and characteristic features, case study

Unit 3
Title – Distributed Resource Management (08 Hrs)
U3.1.

DISTRIBUTED FILE SYSTEMS – Introduction, Distributed File System Design (The File Service Interface, The Directory Server Interface, Semantics of File Sharing), Distributed File System Implementation (File Usage, System Structure, Caching, Replication- Distributed Shared Memory – Algorithms for Implementing
DISTRIBUTED SHARED MEMORY– Introduction, general architecture of DSM systems, design and implementation issues of DSM, granularity, structure of shared memory space, consistency models, replacement strategy, thrashing,
LOAD BALACNING and FAULT TOLERENCE: Issues in Load Distributing – Scheduling Algorithms – Synchronous and Asynchronous Check Pointing and Recovery – Fault Tolerance – Two-Phase Commit Protocol – Non-blocking Commit Protocol – Security and Protection.

U3.2. Self Study: Design and Implementation Issues of DSM.

Unit 4
Title – Database Operating System: (06 Hrs)
U4.1

Introduction, Requirements of Database Operating System, Concurrency Control: Database System, Concurrency Model, Problem of Concurrency Control, Serializability Theory, Distributed Database System Concurrency Control Algorithm: Basic Synchronization Primitives, Lock Based Algorithm, Timestamp Based Algorithms

U4.2. Self Study: Optimistic Algorithm, Data Replication

Unit 5
Title- Security and Protection (06 Hrs)
U5.1.

Security and Protection: Security-threats & goals. Penetration attempts, Security Policies & mechanisms, Authentication. Protections & access control Formal models of protection Cryptography, worms & viruses.

U5.2. Self Study: Threats, Attacks, Assets, Intruders, Overview of Malicious Software

Note: Five assignments to be given to the students on self study, comprising of one assignment from each unit.

Text Books:
T1. Advanced Concepts in operating Systems, Mukesh singhal and Niranjan G. Shivaratri, TMH T2. Operating System, H.M. Beitel, Pearsons,

T2. Operating System Concepts & Design, Milan Milenkovic, McGraw Hill Higher Education,1987.

Reference Books:

R1. Applied Operating System Concepts, Wiley, 2000,A. Silberschatz
R2. G. Coulororis, J. Dollimore & T. Kindberg, Distributed Systems: Concepts and Design, Addison-Wesley.
R3. Operating System Principles, Lubemir F Bic and Alan C. Shaw Pearson Education, 2003

Course Outcome:

Upon completion of the course, graduates will be able to –

1. Ability to solve various synchronization and deadlock problems.
2. Analyze the design issues of distributed system and demonstrate the mutual exclusion problems.
3. Identify the different features of distributed resource management protocols.
4. Evaluate the design issues of distributed operating system.
5. Identify security and protection issues of distributed operating system.

COURSE CODE: CS42173 **REF NO: _____**

Distributed Operating System - Tutorial

Credits: 01 **Teaching Scheme: -** Tutorial 01 Hrs/Week

Prerequisites:

1. Operating System.

Objectives:

- To gain knowledge on Distributed operating system concepts that includes architecture, Mutual exclusion algorithms, Deadlock detection algorithms and agreement protocols
- To gain insight on to the distributed resource management components. The algorithms for implementation of distributed shared memory, recovery and commit protocols

Course Details:

List of Contents

Problems and exercises to be taken up in the following topics:

Tutorial No. 1: Basics of Process synchronization

Tutorial No. 2: Deadlock in Distributed system.

Tutorial No. 3: Basics of distributed Mutual exclusion.

Tutorial No. 4: Election Algorithm.

Tutorial No. 5: Understanding of Distributed File System Design.

Tutorial No. 6: Overview of Distributed Shared memory.

Tutorial No. 7: Concepts of Load balancing algorithm and fault tolerance.

Tutorial No. 8: Understanding of Database Operating System and distributed Concurrency Control.

Tutorial No. 9: Understanding Security-threats and possible protection mechanism

Tutorial No. 10: Overview of cryptography. ,

Text Books:

T1. Advanced Concepts in operating Systems, Mukesh singhal and Niranjan G. Shivaratri, TMH T2. Operating System, H.M. Beitel, Pearsons,
T3. Operating System Concepts & Design, Milan Milenkovic, McGraw Hill Higher Education,1987.

Reference Books:

R1. Applied Operating System Concepts, Wiley, 2000,A. Silberschatz

R2. G. Coulororis, J. Dollimore & T. Kindberg, Distributed Systems: Concepts and Design, Addison-Wesley.
R3. Operating System Principles, Lubemir F Bic and Alan C. Shaw Pearson Education, 2003

Course Outcome:

Upon completion of the course, graduates will be able to –

1. Ability to solve various synchronization and deadlock problems.
2. Analyze the design issues of distributed system and demonstrate the mutual exclusion problems.
3. Identify the different features of distributed resource management protocols.
4. Evaluate the design issues of distributed operating system.
5. Identify security and protection issues of distributed operating system.

COURSE CODE: IT42182 **REF NO:** **To be filled by CD office**

Cryptography and Network Security

Credits: 3 **Teaching Scheme: -** Theory 3 Hrs/Week

Prerequisites:
1. Knowledge on Computer programming
2. Algorithm design
3. Modern algebra (sets, groups etc) and number theory

Objectives:

1. To let the students understand the areas of cryptography and cryptanalysis.
2. To develop a basic understanding of the algorithms used to protect users online and to understand some of the design choices behind these algorithms.
3. To develop a workable knowledge of the mathematics used in cryptology

Course Details:

Unit 1
Title- Introduction to Information Security

(06 Hrs)

U1.1
Introduction to Information Security: Security Goals, Attacks, Security Services and Mechanisms,

Mathematical Background: Integer and Modular Arithmetic, Matrices, Linear Congruence. Groups, Rings, and Fields, GF(p), Euclidean and Extended Euclidean Algorithms, Polynomial Arithmetic, GF(2n). Random Number Generation, Prime Numbers, Fermat's and Euler's Theorems, Primality Testing Methods, Factorization, Chinese Remainder Theorem, Quadratic Congruence, Discrete Logarithms.

U1.2. Self Study: Problem solving on Fermat's & Euler's Theorem, Chinese Remainder Theorem

Unit 2
Title – Traditional Encryption Methods: **(06Hrs)**
U2.1
Symmetric Cipher Model, Substitution Ciphers,
Transposition Ciphers, Block and Stream Ciphers, Rotor Cipher, Steganography.
Symmetric Key Ciphers: Data Encryption Standard, Advanced Encryption Standard.

U2.2. Self Study: Do Coding practices to implement DES algorithm

Unit 3
Title – Asymmetric Key Ciphers: **(06 Hrs)**

U3.1

RSA Cryptosystem, ElGamal Cryptosystem, Elliptic Curve Cryptosystem.

Message Integrity, Authentication: Message Integrity, Random Oracle Model, Message Authentication, MAC Algorithms. Cryptographic Hash Functions: MD Hash Family, Whirlpool, Secure Hash Algorithm. Digital Signature and Authentication: Digital Signature Schemes, Variations and Applications, Entity Authentication.Key Management: Diffie-Hellman Key Exchange.

U3.2. Self Study: A case study on the use of different Asymmetric methods in different systems

Unit 4

Title – Network Security: **(06 Hrs)**

U4.1

Security at the Application Layer: e-mail security, PGP and S/MIME. Security at the Transport Layer: Secure Socket Layer (SSL) and Transport Layer Security (TLS). Security at the Network Layer: IP Security.

U4.2. Self Study: To explore the available open source security testing tools for web applications.

Unit 5

Title- System Security & Internetwork Security **(06 Hrs)**

U5.1

System Security: Malicious Software, Malicious Programs, Viruses, Worms, Malware, Intrusion Detection System, Firewalls.

Internetwork Security: Cloud Security, Introduction to Blockchain and Bitcoin.

U5.2. Self Study: Explore the security challenges lying in modern network computing paradigms like Cloud Computing, Edge Computing, Fog Computing, Mist Computing etc.

Text Books:

1. B. A. Forouzan & D Mukhopadhyay ,Cryptography and Network Security.,McGraw Hill, 2nd ed.2010
2. Cryptography and Network Security: Principles and Practice (7th Edition) 7th Edition, by William Stallings, Pearson, ISBN-13: 978-0134444284, ISBN-10: 0134444280

Reference Books:

1. Cryptography: Theory and Practice (D. R. Stinson) :
2. The Handbook of Applied Cryptography (by Menezes, van Oorschot and Vanstone)
3. Cryptography Engineering(Niels Ferguson, Bruce Schneier, Tadoyoshi Kohno)
4. Applied Cryptography (Bruce Schneier)
5. Modern Cryptography : Theory and Practice (Wenbo Mao)
6. An Introduction to Mathematical Cryptography (J. Hoffstein)

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Identify different security goals and security services.
2. Apply the fundamental mathematical concepts used in security algorithms.
3. Learn and solve problems based on Traditional encryption methods.
4. Gain idea about Asymmetric Key crypto systems.
5. Get familiarize with network security vulnerabilities and their solutions
6. Analyze the vulnerabilities in Systems concerned with System and internetwork security.

COURSE CODE: IT42182 **REF NO:** **To be filled by CD office**

Cryptography and Network Security – Tutorial

Credits:01 **Teaching Scheme: -** Tutorial 01 Hrs/Week

Prerequisites:
1. Knowledge on Computer programming
2. Algorithm design
3. Modern algebra (sets, groups etc) and number theory

Objectives:

1. To let the students understand the areas of cryptography and cryptanalysis.
2. To develop a basic understanding of the algorithms used to protect users online and to understand some of the design choices behind these algorithms.
3. To develop a workable knowledge of the mathematics used in cryptology

Course Details

Problem Solving and Discussion on:

Tutorial 1- Integer and Modular Arithmetic
Tutorial 2- Primality Testing methods
Tutorial 3- Substitution and Transposition Cipher techniques
Tutorial 4- DES and AES algorithm
Tutorial 5- RSA, Elgamal and ECC cryptosystems
Tutorial 6- Key Management techniques
Tutorial 7- SSL protocols & TLS protocols
Tutorial 8- Issues in Network layer security
Tutorial 9- IDS Systems
Tutorial 10-Block chain coding

Text Books:

1. B. A. Forouzan & D Mukhopadhyay ,Cryptography and Network Security.,McGraw Hill, 2nd ed.2010
2. Cryptography and Network Security: Principles and Practice (7th Edition) 7th Edition, by William Stallings, Pearson, ISBN-13: 978-0134444284, ISBN-10: 0134444280

Reference Books:

1. Cryptography: Theory and Practice (D. R. Stinson) :
2. The Handbook of Applied Cryptography (by Menezes, van Oorschot and Vanstone)
3. Cryptography Engineering(Niels Ferguson, Bruce Schneier, Tadoyoshi Kohno)
4. Applied Cryptography (Bruce Schneier)
5. Modern Cryptography : Theory and Practice (Wenbo Mao)
6. An Introduction to Mathematical Cryptography (J. Hoffstein)

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Identify different security goals and security services.
2. Apply the fundamental mathematical concepts used in security algorithms.
3. Learn and solve problems based on Traditional encryption methods.
4. Gain idea about Asymmetric Key crypto systems.
5. Get familiarize with network security vulnerabilities and their solutions.
6. Analyze the vulnerabilities in Systems concerned with System and internetwork security.

COURSE CODE: IT42177 **REF NO: To be filled by CD office**

Parallel and Distributed Systems

Credits: 3 **Teaching Scheme: -** Theory 3 Hrs/Week

Prerequisites:
1. Computer Organization
2. Data Communications and Computer Networks
3. Operating Systems
4. Computer Programming

Objectives:

1. To develop and apply knowledge of parallel and distributed computing techniques and methodologies.
2. To gain experience in the design, development, and performance analysis of parallel and distributed applications.
3. To gain experience in the application of fundamental Computer Science methods and algorithms in the development of parallel applications.
4. To understand the design, testing, and performance analysis of a software system, and to be able to communicate that design to others.

Course Details:

Unit 1
Title- Introduction to Parallel Computing: (06 Hrs)
U1.1.
Parallel programming platforms: Trends in microprocessor architectures, Limitations of memory system performance, Dichotomy of parallel computing platforms, physical organization of parallel platforms, communication costs in parallel machines.
Routing mechanisms for interconnection network, Impact of process processors mapping and mapping techniques.

U1.2. Self Study: Pipelining and Superscalar Architecture. Flynn's Classification. Cache coherence problem.

Unit 2
Title – Parallel Algorithm Design: (06Hrs)

U2.1.
Principles of parallel algorithm design: Decomposition techniques, Characteristics of tasks and interactions, mapping techniques for load balancing, parallel algorithm- models like, Data parallel model, Task graph model, Work pool model, Master slave model , Producer consumer or pipeline model, Hybrid model.

U2.2. Self Study: Methods for containing interactions overheads.

Unit 3
Title – Basic communication operations (08 Hrs)
U3.1

Introduction, One-to-All Broadcast and All-to-One Reduction, All-to-All broadcast and reduction, All-Reduce and prefix sum operations, scatter and gather, All-to-All personalized communication, circular shift.

U3.2. Self Study: Methods of improving the speed of some communication operations.

Unit 4
Title – Analytical modeling of parallel programs: (06 Hrs)
U4.1
Performance metrics for parallel systems, Effect of granularity of performance, Scalability of parallel system, Minimum execution time and minimum cost-optimal execution time, asymptotic analysis of parallel programs, other scalability metrics.

U4.2. Self Study: Comparison between best known sequential algorithm and parallel algorithm in terms of different performance metrics.

Unit 5
Title- Message Passing Paradigm (08 Hrs)
U5.1.
Principle of message – Passing programming, send and receive operations, The Message Passing Interface, Topologies and embedding, Overlapping communication with computation, collective communication and computation operations, Groups and communicators. Dense matrix algorithm: Matrix-vector multiplication, Matrix-matrix algorithm: DNS Algorithm, Canon's Algorithm.

U5.2. Distributed System: Distributed Operating System and Parallel Virtual Machine(PVM)

U5.3. Self Study: Sorting Algorithms.

Text Books:
T1. "Introduction to Parallel Computing", Second Edition, Ananth Gram, Anshul Gupta, George Karypis, Vipin Kumar, Pearson Education.
T2. "Parallel computing Theory and Practice", Second Edition, Michael J. Quinn, TMH.
T3. Andrew S. Tanenbaum, Pearson Education India, 1995 - Distributed operating systems (Computers)

Reference Books:
R1. "Parallel and Distributed Systems", Arun Kulkarni, Nupur Prasad Giri, Nikhilesh Joshi, Bhushan Jadhav, Wiley India Private Limited.
R2. "Using MPI: Portable Parallel Programming with the Message-Passing Interface", William Gropp, Ewing Lusk, Anthony Skjellum, 3rd Edition, MIT Press.
R3. "programming Massively Parallel Processors A Hands on Approach", David B. Kirk and Wen-mei W. Hwu , Morgan Keifmann, Elsevier.

Course Outcomes:
Upon completion of the course, graduates will be able to –

1. Design and analyze parallel computing architecture.
2. Describe the various design issues in a parallel algorithms.

3. Evaluate the impact of interconnection network on parallel/distributed algorithms.
4. Analyze parallel and distributed algorithms in problem solving and apply performance metrics.
5. Describe the different principles of message passing programming, and study the behavior of parallel programs.

COURSE CODE: IT42177 **REF NO: To be filled by CD office**

Parallel and Distributed Systems - Tutorial

Credits: 3 **Teaching Scheme: -** Theory 3 Hrs/Week

Prerequisites:

1. Computer Organization
2. Data Communications and Computer Networks
3. Operating Systems
4. Computer Programming

Objectives:

1. To develop and apply knowledge of parallel and distributed computing techniques and methodologies.
2. To gain experience in the design, development, and performance analysis of parallel and distributed applications.
3. To gain experience in the application of fundamental Computer Science methods and algorithms in the development of parallel applications.
4. To understand the design, testing, and performance analysis of a software system, and to be able to communicate that design to others.

Course Details:

List of Contents

Tutorial No. 1: Understanding Parallel programming platforms
Tutorial No. 2: Problem solving on physical organization of parallel platforms
Tutorial No. 3: Understanding the principles of parallel algorithm design
Tutorial No. 4: Problem solving on task dependency & decomposition
Tutorial No. 5: Understanding basic communication operations i.e. broadcast & reduction
Tutorial No. 6: Problem solving based on different communication operations applied on different networks such as ring, mesh or hypercube.
Tutorial No. 7: Understanding asymptotic analysis of Parallel Programs
Tutorial No. 8: Problem solving on speed up (Amdahl's law, super linear speed up)
Tutorial No. 9: Understanding message passing paradigm
Tutorial No. 10: Problem solving on message transfer protocols and other communication operations.

Text Books:

T1. "Introduction to Parallel Computing", Second Edition, Ananth Gram, Anshul Gupta, George Karypis, Vipin Kumar, Pearson Education.
T2. "Parallel computing Theory and Practice", Second Edition, Michael J. Quinn, TMH.
T3. "programming Massively Parallel Processors A Hands on Approach", David B. Kirk and Wen-mei W. Hwu , Morgan Keifmann, Elsevier.

Reference Books:

R1. "Parallel and Distributed Systems", Arun Kulkarni, Nupur Prasad Giri, Nikhilesh Joshi, Bhushan Jadhav, Wiley India Private Limited.
R2. "Using MPI: Portable Parallel Programming with the Message-Passing Interface", William Gropp, Ewing Lusk, Anthony Skjellum, 3rd Edition, MIT Press.

Course Outcomes:
Upon completion of the course, graduates will be able to –

1. Design and analyze parallel computing architecture.
2. Describe the various design issues in a parallel algorithms.
3. Evaluate the impact of interconnection network on parallel/distributed algorithms.
4. Analyze parallel and distributed algorithms in problem solving and apply performance metrics.
5. Describe the different principles of message passing programming, and study the behavior of parallel programs.

COURSE CODE: MB41301 **REF NO: _____**

Ethics for Engineers Lab

Credits: 01 **Teaching Scheme: -** Laboratory 2 Hrs/Week

Prerequisites: Philosophy of Engineering

Course Objectives:

- To create awareness on Professional Ethics and Human Values.
- To provide basic knowledge about Morality, Ethical Dilemmas, and Professional Virtues.
- To create awareness about code of ethics and industrial standards.
- To inculcate knowledge about social aspects of Business and Environment.

Lab 1: Introduction to Professional Ethics – Nature, Scope and Importance: Discussion & Case Study 1 (2hrs)

Lab 2: Ethical Dilemma and Problems: Discussion & Case Study 2 (2hrs)

Lab 3: Ethics in Marketing (Pricing and Advertising): Discussion & Case Study 3 (2hrs)

Lab 4: Ethics in Finance (Insider Trading, Green Mail, Golden Parachute): Discussion & Case Study 4 (2hrs)

Lab 5: Ethics in Human Resource Management (Worker Rights and Duties, Workplace Safety, Sexual Harassment and Whistle Blowing): Discussion & Case Study 5 (2hrs)

Lab 6: Ethics in Engineering and Technology (Accuracy, Privacy, Property, Accessibility): Discussion & Case Study 6 (2hrs)

Lab 7: Ethical Issues in Society (Black Marketing, Bribery and Corruption): Discussion & Case Study 7 (2hrs)

Lab 8: Ethics in Corporate Social Responsibility: Discussion & Case Study 8 (2hrs)

Lab 9: Role of Ethical Codes and their implementation: Discussion & Case Study 9 (2hrs)

Lab 10: Ethical Practices in different organizations: Discussion & Case Study 10 (2hrs)

Course Outcome:

After completion of the course the students will:

CO1. Understand various social issues, industrial standards and importance of ethics in the engineering domain.

CO2. Aware of ethical responsibilities of an engineer.

CO3. Acquire professional behavior that requires adherence to the highest principles of ethical conduct.

Text Book:

1. Business Ethics – Text & Cases, C.S.V. Murthy, 1st Edition, HPH, 2017.
2. Ethics in Engineering, M.W. Martin, 4th Edition, McGraw Hill

Reference Book:

1. Professional Ethics, R. Subramanian, 2nd Edition, Oxford