



RAMAIAH
Institute of Technology

CURRICULUM

For the Academic year 2018 – 2019

**DEPARTMENT OF INFORMATION
SCIENCE AND ENGINEERING**

V & VI Semester B. E.

RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute, Affiliated to VTU)
BANGALORE – 54

About the Institute

Ramaiah Institute of Technology (RIT) (formerly known as M. S. Ramaiah Institute of Technology) is a self-financing institution established in Bangalore in the year 1962 by the industrialist and philanthropist, Late Dr. M S Ramaiah. The institute is accredited with A grade by NAAC in 2016 and all engineering departments offering bachelor degree programs have been accredited by NBA. RIT is one of the few institutes with prescribed faculty student ratio and achieves excellent academic results. The institute was a participant of the Technical Education Quality Improvement Program (TEQIP), an initiative of the Government of India. All the departments have competent faculty, with 100% of them being postgraduates or doctorates. Some of the distinguished features of RIT are: State of the art laboratories, individual computing facility to all faculty members. All research departments are active with sponsored projects and more than 140 scholars are pursuing PhD. The Centre for Advanced Training and Continuing Education (CATCE), and Entrepreneurship Development Cell (EDC) have been set up on campus. RIT has a strong Placement and Training department with a committed team, a good Mentoring/Proctorial system, a fully equipped Sports department, large air-conditioned library with over 1,00,000 books with subscription to more than 300 International and National Journals. The Digital Library subscribes to several online e-journals like IEEE, JET etc. RIT is a member of DELNET, and AICTE INDEST Consortium. RIT has a modern auditorium, several hi-tech conference halls and all are air-conditioned with video conferencing facilities. It has excellent hostel facilities for boys and girls. RIT Alumni have distinguished themselves by occupying high positions in India and abroad and are in touch with the institute through an active Alumni Association. RIT obtained Academic Autonomy for all its UG and PG programs in the year 2007. As per the National Institutional Ranking Framework, MHRD, Government of India, Ramaiah Institute of Technology has achieved 60th rank in 2018 among the top 100 engineering colleges across India.

About the Department

Information Science and Engineering department is established in the year 1992 with an objective of producing high-quality professionals to meet the demands of the emerging field of Information Science and Engineering. Department also started M.Tech program in Software Engineering in the year 2004 and has been recognized as R&D center by VTU in 2012. The department is accredited by the NBA in 2001, 2004, 2010 and reaccredited in 2015 under Tier-1. Department has highly qualified and motivated faculty members, and well equipped state of the art laboratories. All faculty members are involved in research and technical papers publications in reputed journals, conferences across the world. Strong collaboration with industries and high profile institutions is in place for curriculum updates, more hands on training, practical's, project based learning, expert lectures, partial course deliveries by industry experts and student interns to keep an healthy academic atmosphere. Department is successfully conducting seminars, conferences and workshops for students and academicians in the emerging areas of Information Technology. Some of the laboratories have also been set up in collaboration with industries such as Intel, Microsoft, Apple, SECO, Honeywell, EMC², NVIDIA and IBM. Also, an echo system is built to initiate start-ups at the department level along with the mentorship. All the above potential activities have lead to high profile placements, motivation to become an entrepreneur, and encouragement for higher learning.

VISION OF THE INSTITUTE

To be an Institution of International Eminence, renowned for imparting quality technical education, cutting edge research and innovation to meet global socio economic needs

MISSION OF THE INSTITUTE

MSRIT shall meet the global socio-economic needs through

- Imparting quality technical education by nurturing a conducive learning environment through continuous improvement and customization
- Establishing research clusters in emerging areas in collaboration with globally reputed organizations
- Establishing innovative skills development, techno-entrepreneurial activities and consultancy for socio-economic needs

QUALITY POLICY

We at M. S. Ramaiah Institute of Technology strive to deliver comprehensive, continually enhanced, global quality technical and management education through an established Quality Management System complemented by the synergistic interaction of the stake holders concerned

VISION OF THE DEPARTMENT

To evolve as an outstanding education and research center of Information Technology to create high quality Engineering Professionals for the betterment of Society

MISSION OF THE DEPARTMENT

- To provide a conducive environment that offers well balanced Information Technology education and research.
- To provide training and practical experience in fundamentals and emerging technologies.
- To nurture creativity for overall personality development.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Become competent Information Technology professionals with continuous progress in career or learning.

PEO2: Enhance the skills in developing computing systems using modern tools and technologies.

PEO3: Function effectively as professionals in a team environment or individually.

PROGRAM OUTCOMES (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Problem Solving Skills, ability to understand and analyze the Information Technology problems and develop computer programs.

PSO2: Applied Engineering Skills, ability to apply standard practices and strategies in Software Development.

PSO3: Communication and Higher Learning, ability to exchange knowledge and continue learning advances in the field of Information Technology.

Curriculum Course Credits Distribution

SCHEME OF TEACHING

V SEMESTER

| Sl. No. | Course Code | Course Name | Category | Credits | | | | | Contact Hours |
|---------|-------------|--|----------|---------|---|---|---|-------|---------------|
| | | | | L | T | P | S | Total | |
| 1. | IS51 | Computer Networks | PC-C | 4 | 0 | 0 | 0 | 04 | 04 |
| 2. | IS52 | Operations Research | PC-C | 3 | 1 | 0 | 0 | 04 | 05 |
| 3. | IS53 | Operating Systems | PC-C | 3 | 0 | 0 | 0 | 03 | 03 |
| 4. | IS54 | Database Management Systems | PC-C | 3 | 0 | 0 | 1 | 04 | 03 |
| 5. | IS55 | Intellectual Property Rights | HSS | 2 | 0 | 0 | 0 | 02 | 02 |
| 6. | ISL56 | Computer Networks Laboratory | PC-C | 0 | 0 | 1 | 0 | 01 | 02 |
| 7. | ISL57 | Database Management Systems Laboratory | PC-C | 0 | 0 | 1 | 0 | 01 | 02 |
| 8. | ISL58 | Scripting Languages Laboratory | PC-C | 0 | 0 | 2 | 0 | 02 | 04 |
| 9. | ISEAX | Elective A | PC-E | 4 | 0 | 0 | 0 | 04 | 04 |
| Total | | | | 19 | 1 | 4 | 1 | 25 | 29 |

Elective A:

| | |
|-------|-----------------------------|
| ISEA1 | Digital Image Processing |
| ISEA2 | Natural Language Processing |
| ISEA3 | Embedded Computing Systems |

VI SEMESTER

| Sl. No. | Course Code | Course Name | Category | Credits | | | | | Contact Hours |
|---------|-----------------|---|----------|---------|---|---|---|-------|---------------|
| | | | | L | T | P | S | Total | |
| 1. | IS61 | System Software | PC-C | 4 | 0 | 0 | 0 | 04 | 04 |
| 2. | IS62A/ IS62B | Object Oriented Analysis and Design Patterns using C++/JAVA | PC-C | 3 | 0 | 0 | 1 | 04 | 03 |
| 3. | IS63 | Java and J2EE | PC-C | 3 | 0 | 0 | 1 | 04 | 03 |
| 4. | ISL64 | Mini-Project | PW | 0 | 0 | 6 | 0 | 06 | 12 |
| 5. | ISL65 | System Software Laboratory | PC-C | 0 | 0 | 1 | 0 | 01 | 02 |
| 6. | ISL66 | Object Oriented Analysis and Design Patterns Laboratory | PC-C | 0 | 0 | 1 | 0 | 01 | 02 |
| 7. | ISL67 | Java and J2EE Laboratory | PC-C | 0 | 0 | 1 | 0 | 01 | 02 |
| 8. | ISEBX | Elective B | PC-E | 4 | 0 | 0 | 0 | 04 | 04 |
| Total | | | | 14 | 0 | 9 | 2 | 25 | 32 |

Elective B:

| | |
|-------|----------------------------------|
| ISEB1 | Machine Learning |
| ISEB2 | Compiler Design |
| ISEB3 | Distributed Storage Technologies |

COMPUTER NETWORKS

Course Code: IS51

Course Coordinator: Suresh Kumar K R

Credits: 4:0:0:0

Contact Hours: 56L

UNIT-I

Networklayer:Logicaladdressing - IPV4addresses, Address space, Notations, Classful addressing, Classless addressing, Network Address Translation;IPV6addresses, Structure and Address Space.**Networklayer:Internetprotocol**-IPV4 datagram, fragmentation, checksum, options; IPV6 packet format, advantages, extension headers;TransitionfromIPV4toIPV6.

UNIT-II

Addressmapping,Errorreporting,&Multicasting-Addressmapping, Logical to physical address - ARP, Physical to Logical address - RARP, BOOTP and DHCP; Internet Control Message Protocol (ICMP), Internet Group Management Protocol (IGMP). **Network layer: Delivery, Forwarding, & Routing** – Direct Vs Indirect delivery, Forwarding Techniques, Forwarding Process, Routing Table; **Unicast Routing Protocols** – Optimization, Intra and Inter Domain Routing, Distance Vector Routing, Link State Rout Routing, Path Vector Routing.

UNIT-III

Multicast Routing Protocols – Introduction to Unicast, Multicast, Broadcast and Anycast, applications, Unicast Routing Vs Multicast Routing, Source Based Tree Routing, Group Shared Tree Routing, Multicast Link State Routing, Multicast Distance Vector, CBT, MBONE. **Transport Layer** - Process-to-Process Delivery, User Datagram Protocol, Transmission Control Protocol, SCTP-services and features.

UNIT-IV

Congestion control & QOS - Data traffic, Congestion, Congestion control, Two Examples – Congestion control in TCP and Frame Relay, Quality of Service, Techniques to improve QOS. **Application Layer: Domain Name System** -Namespace, Domain name space, Distribution of Name space, DNS in internet, Resolution. **Case study:** QoS Requirements In Portable Devices.

UNIT-V

Remote logging – TELNET; **Electronic mail** – Architecture, User Agent, Message Transfer Agent: SMTP; **File transfer** - File transfer protocol (FTP). **Network Management: SNMP** - Network management system; Simple Network Management Protocol – concept, management components, structure of management information, Management information base, Lexicographic ordering, SNMP PDUs. **Case study:** Mobile Networks Applications.

Text Book:

1. Behrouz A. Forouzan, Data Communications and Networking, Fourth Edition, Tata McGraw-Hill, 2006.

References:

1. Alberto Leon-Garcia and Indra Widjaja, Communication Networks –Fundamental Concepts and Key architectures, Second Edition, Tata McGraw-Hill, 2004.
2. William Stallings, Data and Computer Communication, Eight Edition, Pearson Education, 2007.
3. Larry L. Peterson and Bruce S. David, Computer Networks – A Systems Approach, Fourth Edition, Elsevier, 2007.

Course Outcomes:

1. Identify and solve the problems associated with transition from IPV4 to IPV6. (PO-1,2,3) (PSO-1,2)
2. Use different protocols to achieve Address mapping, Error reporting and unicast routing. (PO-1,2,3)(PSO-1)
3. Paraphrase multicast routing and distinguish between different transport layer protocols.(PO- 1, 2) (PSO-2)
4. Analyze different techniques to improve QoS and illustrate domain namespace resolution.(PO- 1,2,3)(PSO-2, 3)
5. Illustrate the working of various application layer services like Remote login, E-mail, FTP, SNMP and simple network management protocol. (PO- 2, 12) (PSO-2, 3)

OPERATIONS RESEARCH

Course Code: IS52

Course Coordinator: Myna A N

Credits: 3:1:0:0

Contact Hours: 42L+14T=56

UNIT-I

Introduction to Operations Research (OR) and Linear Programming (LP): OR Models, Solving the OR Model, Phases of an OR Study; Modeling with Linear Programming (LP), Two-variable LP Model, Graphical LP solution, Solution of a Maximization/Minimization Model; Computer solution with Excel Solver; LP Model in Equation Form, Transition from Graphical to Algebraic Solution; The Simplex Method; TORA Tool

UNIT-II

The Simplex Method: Special cases in the Simplex Method, Degeneracy, Alternative optima, Unbounded solutions, Non-existing solutions; Sensitivity Analysis (SA); Graphical SA, Artificial Variable Techniques: Two Phase Method, Big-M method; TORA Tool

UNIT-III

Duality and Game Theory: Introduction: Concept of Duality; Definition of Primal-Dual Problems; General Rules for Converting any Primal into its Dual; Introduction to Dual Simplex Method; Computational Procedure of Dual Simplex Method; Illustrative Examples. Introduction of Game Theory, Characteristics of Games Theory; Minimax (Maximin) Criterion and Optimal Strategy; Saddle Point, Optimal Strategies and the value of game; Solution of Games with Saddle Point(s); Rectangular Games without Saddle Point; Arithmetic Method for (2x2) Games, Gambit tool for Game theory.

UNIT-IV

Project Management by PERT-CPM: Introduction, Applications of PERT/CPM Techniques, Basic Steps in PERT/CPM Techniques; Network Diagram Representation, Rules for Drawing Network Diagram, Labelling Fulkerson's 'I-J' Rule; Time Estimates and Critical Path in Network Analysis; Optimum Duration and Minimum Duration Cost; Project Evaluation and Review Technique, Updating, Resource Allocation, TORA tool.

UNIT-V

Transportation and Assignment Problems: Mathematical Formulation of transportation problem (TP); Matrix Form of TP; Feasible Solution, Basic Feasible Solution and Optimum Solution; Tabular Representation; Special Structure of Transportation Table and their Problems; Initial Basic Feasible Solution to TP; Moving Towards Optimality; Degeneracy in TP; Unbalanced TP; Mathematical Formulation of Assignment Problem (AP); Fundamental Theorems; Hungarian Method for AP; TORA tool.

Tutorial Exercises:

1. Problems on Formulation of Linear Programming Problems,
2. Problems on Graphical Solutions to Linear Programming Problems
3. Problems on Simplex Method
4. Problems on special cases in simplex method
5. Problems on Two Phase Method
6. Problems on Big-M Method
7. Problems on Dual Simplex Method
8. Problems on CPM
9. Problems on PERT
10. Problems on Game theory
11. Problems on Transportation Problems
12. Problems on Assignment Problems

Text Books:

- 1.Operations Research: An Introduction – Hamdy A Taha – 9th Edition, Pearson Education, India, 2011.
- 2.Operations Research – S.D. Sharma – 16th Edition, KNRN Publications, 2009.

Reference:

- 1.Introduction to Operations Research – Frederick S. Hillier and Gerald J. Lieberman – 9th Edition, Tata McGraw Hill, 2009

Course Outcomes:

1. Formulate and solve linear programming model using graphical method.(PO-1, 2,3,5) (PSO- 1, 2)
2. Solve a linear programming model using simplex methods (PO-1, 2, 3, 5) (PSO-1, 2)
3. Apply game theory to model, analyze and solve real-world problems. (PO-1, 2, 3, 5) (PSO-1, 2)
4. Analyze a project network using PERT and CPM techniques. (PO-1, 2, 3, 5) (PSO-1, 2)
5. Solve Transportation and assignment problems. (PO-1, 2, 3, 5) (PSO-1, 2)

OPERATING SYSTEMS

Course Code: IS53

Course Coordinator: Pushpalatha M N

Credits: 3:0:0:0

Contact Hours: 42L

UNIT-I

Introduction: Operating System Structure, Operating System Operations, Process Management, Memory Management, Storage Management, Protection and Security; Process Concept: Process Scheduling, Operations on Processes, Interprocess Communication; Process Scheduling: Scheduling Criteria, Scheduling algorithms Multiple processor scheduler.

UNIT-II

Synchronization: The Critical Section Problem, Peterson's Solution, Synchronization Scheduling Algorithms, Multiple-Processor Scheduling Hardware, Semaphores, Classical Problems of Synchronization, Monitors, Synchronization Examples; Deadlocks: System Model, Deadlock Characterization, Methods for handling Deadlocks, Deadlock Prevention, Deadlock Avoidance and detection, Recovery from Deadlock.

UNIT-III

Memory Management Strategies: Background, Swapping, Contiguous Memory Allocation, Paging, Structure of Page Table, Segmentation; Virtual Memory Management: Background, Demand Paging, Copy on Write, Page Replacement, Allocation of frames, Allocating Kernel Memory.

UNIT-IV

File System: File Concept, Access Methods, Directory Structure, Protection; Implementing File Systems: File System Structure, File System Implementation, Directory Implementation, Allocation Methods, Free Space Management, Efficiency and Performance, Recovery; Secondary-Storage Structure: Disk structure, Disk Attachment, Disk Scheduling Methods, Disk management, Swap-Space Management.

UNIT-V

System Protection: Goals of Protection, Principle of Protection, Access Matrix, Domain of Protection, Access Matrix and its implementation, Case Study on The Linux System: Design Principles, Kernel Modules, Process Management, Scheduling, Memory Management File Systems, Interprocess Communication.

Text Book:

1. Silberschatz, Galvin, Gagne, Operating System Concepts, 7/e, John Wiley and sons, 2007.

References:

1. Dhamdhere D. M., Operating Systems, A Concept Based Approach, 2/e, Tata McGraw Hill, 2006.
2. Tanenbaum A. S., Modern Operating Systems, 3/e, Pearson Education, 2008.

Course Outcomes:

1. Discuss operating system concepts and analyze CPU scheduling algorithms. (PO- 1,2,3) (PSO-1,2)
2. Describe the various methods for accessing shared data and handling deadlocks. (PO-1,2,3) (PSO-1,2)
3. Employ the techniques for memory management during process execution. (PO-1,2,3) (PSO-1,2)
4. Describe the file system and secondary storage structure.(PO-1,2,3) (PSO-1,2)
5. Articulate the need for information protection mechanisms in OS and illustrate the working of modern operating system. (PO –1,2) (PSO-1,2)

DATABASE MANAGEMENT SYSTEMS

Course Code: IS54

Course Coordinator: Prashanth Kambli

Credits: 3:0:0:1

Contact Hours: 42L

UNIT-I

Introduction: Characteristics of Database approach; Advantages of using DBMS approach, Data models, Schemas and Instances; **Three-schema architecture and data independence:** Database languages and interfaces; Classification of Database Management systems; Using High-Level Conceptual Data Models for Database Design;

An Example Database Application: University Database, Railway Management database, Airways database

Self-Study: MongoDB –Overview, Advantages, Environment,

UNIT-II

Entity Types, Entity Sets, Attributes and Keys; Relationship types, Relationship Sets, Roles and Structural Constraints; **Weak Entity Types;** **Refining the ER Design;** ER Diagrams, Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; **Update Operations,** Transactions and dealing with constraint violations.

Self-Study: MongoDB: Data Modelling, Create database, Drop database, create collection, Drop collection Alter

UNIT-III

JOINS

Unary Relational Operations: SELECT and PROJECT; Relational Algebra Operations from Set Theory; Binary Relational Operations: JOIN and DIVISION; **Additional Relational Operations,** Relational Database Design Using ER- to-Relational Mapping, Informal Design Guidelines for Relation Schemas; Functional Dependencies; **Normal Forms Based on Primary Keys-1NF, 2NF, 3NF, BCNF.**

Self-Study: MongoDB: Data Types, Insert, Update, Querying Database, Projection, Sorting Database.

UNIT-IV

SQL Data Definition and Data Types; Specifying basic constraints in SQL; Schema change statements in SQL; Basic queries in SQL; **More complex SQL Queries.** Insert, Delete and Update statements in SQL; Views (Virtual Tables) in SQL. **PL/SQL:** Data types in PL/SQL, Control statements (IF, LOOP, WHILE, FOR), **Cursor,** Passing Parameter to Cursor, Implicit Cursor, Exceptions, Procedures and functions.

Self-Study: MongoDB: Aggregation, Replication.

UNIT-V

Introduction to Transaction Processing, Transaction and System Concepts, Desirable properties of Transactions, Characterizing Schedules Based on Recoverability, Characterizing Schedules Based on Serializability, Transaction Support in SQL; **Two-phase Locking Techniques for concurrency Control,** Concurrency Control based on Timestamp Ordering, Recovery concepts, Recovery techniques based on Deferred Update, Recovery techniques based on Immediate Update, Shadow paging, The ARIES recovery algorithm.

Self-Study: MongoDB: Sharding and Create Backup

Text Books:

1. Ramez Elmasri and Shamkant B.Navarthe: Fundamentals of Database Systems, 5th Edition, Addison-Wesley, 2007.
2. Benjamin Rosenzweig, Elena Silvestrova Rakhimov : Oracle PL/SQL by Example, 4th Edition, 2010.
3. <https://www.tutorialspoint.com/mongodb/mongodbTutorial.pdf>, <http://www.j2eebrain.com/mongodb-tutorial>

References:

1. Silberschatz, Korth and Sudharshan: “Data base System Concepts, Fifth Edition, Mc-GrawHill, 2006”.
2. Raghu Ramakrishnan, Johannes Gehrke, “Database Management Systems, Third Edition”, McGraw-Hill, 2003.
3. Kyle Banker Peter Bakkum Shaun Verch Douglas Garrett Tim Hawkins
“MongoDB Action 2nd Edition”

Course Outcomes:

1. Utilize the basic Database concepts and design ER model for any Database Application. (PO- 1, 2, 3, 9, 10, 12) (PSO- 2, 3)
2. Convert ER model to Relational model. (PO- 3, 9, 10, 12) (PSO- 1, 2 ,3)
3. Design a relational database schema by applying normalization procedure for any database application (PO- 1, 3, 9, 10, 12) (PSO-1,2,3)
4. Design and develop a real world database application (PO- 1, 3, 9, 10, 12) (PSO-1,2,3)
5. Utilize the concepts of Transaction processing, concurrency control, recovery techniques to appreciate the database transactions. (PO- 1, 12) (PSO- 2)

INTELLECTUAL PROPERTY RIGHTS

Course Code: IS55

Course Coordinator: Naresh E

Credits: 2:0:0:0

Contact Hours: 28L

UNIT-I

Introduction: Meaning, Relevance, Business Impact, Protection of Intellectual Property, Copyrights, Trademarks, Patents, Designs, Utility Models, Trade Secrets and Geographical Indications Bio-diversity and IPR Competing Rationales for Protection of Intellectual Property Rights, Introduction to the leading International Instruments concerning Intellectual Property Rights: the Berne Convention, Universal Copyright Convention, The Paris Convention, Patent Co-operation Treaty, TRIPS, The World Intellectual Property Organization(WIPO).

UNIT-II

Patents: Concept of Patent, Product / Process Patents & Terminology, Duration of patents Law and Policy Consideration Elements of Patentability - Novelty and Non Obviousness (Inventive Steps and Industrial Application, Non- Patentable Subject Matter Procedure for Filing of Patent Application and types of Applications, Procedure for Opposition, Revocation of Patents, Ownership and Maintenance of Patents, Assignment and licensing of Patents, Working of Patents- Compulsory Licensing, Patent Database and Information Systems, Preparation of Patent documents, Process for Examination of Patent Application, Patent Infringement

UNIT-III

Trademarks - The rationale of protection of trademark for commercial and of consumer rights , Definition and concept of Trademarks , Different kinds of marks (brand names, logos, signatures, symbols, well known marks, certification marks and service marks), Non Registrable Trademarks Procedure for Registration of Trademarks, Opposition Procedure, Assignment/Transmission / Licensing of Trademarks, Infringement of Trademarks.

UNIT-IV

Copyright- Nature of Copyright, Works in which Copyrights subsist, Author & Ownership of Copyright Rights Conferred by Copyright Assignment, Transmission, Licensing of Copyrights Infringement of Copyrights Remedies & Actions for Infringement of Copyrights Copyright Societies, Office, Board, Registration of Copyrights & Appeals International Conventions Copyright pertaining to Software/Internet and other Digital Media Remedies.

UNIT-V

IPR in the Field of Information Technology. Recent Developments in Patent System, Subject matter of software design law definition, subject matter Law relating to embedded software design and registration in India, Infringement of software design rights. Technological and Legal Developments in Intellectual Property, IP in Cyberspace.

Text Books:

1. Dr. B. L. Wadhera, Law Relating to Intellectual Property, Universal law Publishing Co. Ltd. 2009.
2. Aswani Kumar Bansal, Law of Trademarks in India.

Reference books:

1. Intellectual Property Rights, Handbook/Notes.
2. Course materials for one year P.G. Diploma in IPR from NLSIU, Bangalore by Prof. T. Ramakrishna.
3. Case studies from internet sources.

Course Outcomes:

1. Identify the principles of Intellectual Property Rights and its importance in the field of Information Technology. (PO-6,8)(PSO-3)
2. Analyze the importance of patents, trademarks and copyrights.(PO-6,8)(PSO- 3)
3. Describe the procedure for obtaining a patent, registering a trademark andowning copyrights. (PO-6,8)(PSO-3)

COMPUTER NETWORKS LABORATORY

Course Code: ISL56

Course Coordinator: Sandeep B L

Credits: 0:0:1:0

Contact Hours: 14P

Part A: Implement the following using C/C++:

1. Using TCP/IP sockets, write a client-server program to make client send the file name and the server to send back the contents of the requested file name “sample.txt” with the following contents:”Hello we are at Computer Networks Lab”. Display suitable error message in case the file is not present in the server.
2. Write a program to archive Traffic management at Flow level by implementing Leaky Bucket Algorithm.
3. Given a graph, each node A knows the shortest path to node Z and node A can determine its shortest path to Z by calculating the minimum cost. Now when packet flows through a path it incurs some cost to the network, find shortest paths from src to all nodes in the given graph using Bellman Ford Algorithm. The graph may contain negative weight edges.
4. Given a graph find shortest paths from source to all nodes in the graph using Dijkstra’s shortest path algorithm.
5. Write a program for implementing the error detection technique for data transfer in unreliable network code using CRC (16-bits) Technique.
6. Write a program to implement internet checksum for error correction and detection.
7. Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner. Implement priority queuing as a technique to improve Quality of Service.

Part B: Simulation Using NS-2

1. Simulate three nodes point-to-point networks with duplex links between them. Set the queue size and vary the bandwidth and find the number of packets dropped
2. Simulate the different types of internet traffic such as FTP and TELNET over network and analyze the throughput
3. Simulate a four-node point-to-point network, and connect the links as follows: n0->n2, n1->n2 and n2->n3. Apply TCP agent changing the parameters and determine the number of packets sent/received by TCP/UDP
4. Simulate the transmission of ping messages over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.
5. Simulate an Ethernet LAN using n nodes and set multiple traffic nodes and determine collision across different nodes.
6. Simulate an Ethernet LAN using n nodes (6-10), change error rate and data rate and compare throughput.

Text Book:

1. Behrouz A. Forouzan, Data Communications and Networking, Fourth Edition, Tata McGraw-Hill, 2006.

Course Outcomes:

1. Design and implement the functionalities of various layers of the OSI model. (PO – 3, 4) (PSO – 2)
2. Simulate and analyze the network behavior against various parameters using NS2. (PO – 1, 2, 3, 4, 5) (PSO – 2)

DATABASE MANAGEMENT SYSTEMS LABORATORY

Course Code: ISL57

Course Coordinator: Sunitha R S

Credits: 0:0:1:0

Contact Hours: 14P

The Database Management Systems Lab consists of two components: Part A and Part B.
One exercise from each is compulsory.

Part A

I. Execute the following exercises

1. Consider the relations

EMPLOYEE (SSN, Name, DeptNo)

ASSIGNED_TO (USN, ProjectNo)

PROJECT (ProjectNo, ProjectArea)

Create the above tables, insert suitable tuples and perform the following operations in Oracle SQL

- a. Obtain the SSN of employees assigned to database projects.

- b. Find the number of employees working in each department

- c. Update the ProjectNo of Employee bearing SSN=1 to ProjectNo=20

2. Create the below tables, insert suitable tuples and perform the following operations using MongoDB

EMPLOYEE (SSN, Name, DeptNo)

ASSIGNED_TO (USN, ProjectNo)

- a. List all the employees of department "XYZ".

- b. Name the employees working on Project Number :#PNo

II. Execute the following exercises

1. Consider the relations

PART (PNO, PNAME, COLOUR),

SUPPLIER (SNO, SNAME, ADDRESS)

SUPPLY (PNO, SNO, QUANTITY)

Create the above tables, insert suitable tuples and perform the following operations in Oracle SQL:

- a. Obtain the PNO of parts supplied by supplier 'Ram'.

- b. Obtain the Names of suppliers who supply bolts

- c. Delete the parts which are green in colour

2. Create the below tables, insert suitable tuples and perform the following operations using MongoDB

PART (PNO, PNAME, COLOUR),

SUPPLY (SNO, SNAME, PNO, ADDRESS)

- a. Update the parts identifier

- b. Display all suppliers who supply the part with part identifier: #PNO

III. Execute the following exercises

1. Consider the relations

BOAT (BID, BNAME, COLOUR),

SAILOR (SID, SNAME, AGE, RATING)

RESERVES (BID, SID, DAY)

Create the above tables, insert suitable tuples and perform the following operations in Oracle SQL:

- a. Obtain the bid of the boats reserved by 'Ram'.

- b. Retrieve the bid of the boats reserved by all the sailors.

- c. Find the number of boats reserved by each sailor

2. Create the below tables, insert suitable tuples and perform the following operations using MongoDB

BOAT (BID, BNAME, COLOUR)

RESERVES (BID, SNAME, SID, DAY)

- a. Obtain the number of boats obtained by sailor :#sname

- b. Retrieve boats of color :"White"

IV. Execute the following exercises

1. Consider the relations

PART (PNO, PNAME, COLOUR),

WAREHOUSE (WNO, WNAME, CITY)

SHIPMENT (PNO, WNO, QUANTITY, DATE)

Create the above tables, insert suitable tuples and perform the following operations in Oracle SQL:

- a. Obtain the Names of warehouses which have shipped red coloured parts.
 - b. Retrieve the PNO of the parts shipped by all the warehouses.
 - c. Find the number of parts supplied by each warehouse
2. Create the below tables, insert suitable tuples and perform the following operations using MongoDB
- PART (PNO, PNAME, COLOUR)
- SHIPMENT (PNO, WNO, WNAME, QUANTITY, DATE)
- a. Find the parts shipped from warehouse :Wname"
 - b. List the total quantity supplied from each warehouse

V. Execute the following exercises

1. Consider the relations
- BOOK (ISBN, TITLE, AUTHOR, PUBLISHER)
- STUDENT (USN, NAME, SEM, DEPTNO),
- BORROW (ISBN, USN, DATE)

Create the above tables, insert suitable tuples and perform the following operations in Oracle SQL:

- a. Obtain the name of the student who has borrowed the book bearing ISBN '123'
- b. Obtain the Names of students who have borrowed database books.
- c. Find the number of books borrowed by each student.

2. Create the below tables, insert suitable tuples and perform the following operations using MongoDB
- BOOK (ISBN, TITLE, AUTHOR, PUBLISHER)
- BORROW (ISBN, USN, DATE)
- a. Obtain the name of the student who has borrowed the book bearing ISBN '123'.
 - b. Obtain the Names of students who have borrowed database books.

Part B

Execute the following programs in PL/SQL

Consider the following database for a BANK system

BRANCH (Code, Name, Assets)
 CUSTOMER (SSN, Name, Place)
 ACCOUNT (AccNo, SSN, Code, Balance)

- i) Create the above tables by stating the primary and foreign keys
- ii) Insert the following tuples to the tables

BRANCH

CUSTOMER

| Code | Name | Assets |
|------|------|--------|
| B1 | MSR | 10000 |
| B2 | RNR | 20000 |
| B3 | SMR | 15000 |
| B4 | SKR | 25000 |

| SSN | Name | Place |
|-----|------|-------|
| 1 | Ram | BNG |
| 2 | Asha | MNG |
| 3 | Usha | MYS |
| 4 | Sri | DEL |

ACCOUNT

| AccNo | SSN | Code | Balance |
|-------|-----|------|---------|
| A1 | 1 | B1 | 100000 |
| A2 | 1 | B1 | 200000 |
| A3 | 2 | B2 | 100000 |
| A4 | 3 | B2 | 100000 |
| A5 | 3 | B2 | 100000 |
| A5 | 3 | B2 | 100000 |
| A7 | 4 | B2 | 200000 |

1. Write a PL/SQL program to display the contents of the above tables and then update the balance of a few accounts.
2. Write a program that gives all employees in department 10 a 15% pay increase. Display a message displaying how many employees were awarded the increase.
3. Write a PL/SQL program to check whether a given number is prime or not
4. Using cursors demonstrate the process of copying the contents of one table to a new table
5. Write a PL/SQL program to print the first 8 fibonacci numbers
6. Write a PL/SQL procedure to find the factorial of a given number and a program to call the same
7. Write a PL/SQL program to check whether a given number is palindrome or not.

8. Consider the following relation schema.

EMPLOYEE (SSN, Name, sal, DeptNo)

Write a trigger to raise an error if the table is modified on a specific day (Eg.Saturday or Sunday) of the week.

Text Book:

1. Benjamin Rosenzweig, Elena Silvestrova Rakhimov: Oracle PL/SQL by Example, 4th Edition, 2010.

Course Outcomes:

1. Transform an information model into a relational database schema by using data definition language and/or utilities to implement the schema using a DBMS. (PO - 1, 2) (PSO - 1,2)
2. Formulate, using SQL/MongoDB, solutions to a broad range of query and data update problems. (PO - 1, 2) (PSO - 1,2)
3. Demonstrate a rudimentary understanding of programmatic components for a database such as triggers, Packages and functions. (PO - 1, 2) (PSO - 1,2)

SCRIPTING LANGUAGES LABORATORY

Course Code: ISL58

Course Coordinator: Dr.Mydhili K Nair

Credits: 0:0:2:0

Contact Hours: 28P

PART A

- **Python Introduction** - Simple Problems in Python based on:
1) Selection Constructs 2) Looping Constructs
- **Using Python** - Simple Problems in Python based on:
1) Lists 2) Tuples 3) Dictionary
- **Combining Python Basics and its Data Structures** - Problems in Python based on:
Selection, Looping with Lists, Tuples, Dictionaries
- **Using Python** - Problems in Python based on:
1) Functions 2) Class
- **Python for Data Science** - Data Manipulation on IRIS Data Set
- **Python for Data Science** - Data Visualization based on IRIS Data Set
- **HTML Basics Introduction** - Creating Static Pages with different HTML Components
- **HTML Basics Introduction** - Creating Static Pages with hyperlinks and images
- **Javascript Introduction** - Basics of client side scripting
- **Using Javascript** - HTML along with Javascript on client side
- **Combining HTML & Javascript** - Programs with Client side event handling scenarios
- **Using Javascript** - Loading JSON data dynamically on the client side.

PART B

1. **Introduction to Python:** Write Python programs to do the following:
 - a) Read a list of elements. Create a new list having all the elements minus the duplicates (Use functions). Use one-line comprehensions of create a new list of even numbers. Create another list reversing the elements.
 - b) Write a python program to count the frequency of words in a given file.
 - c) Read a list of numbers. Uses a recursive function to find the maximum of ‘n’ numbers.
2. **Introduction to Python Functions:** Write a temperature converter python program, which is menu driven. Each such conversion logic should be defined in separate functions. The program should call the respective function based on the user’s requirement. The program should run as long as the user wishes so. Provide an option to view the conversions stored as list of tuples with attributes - from unit value, to unit value sorted by the user’s choice (from-value or to-value).
3. **Python Classes:** Write a python class to reverse a sentence (initialized via constructor) word by word. Example: “I am here” should be reversed as “here am I”. Create instances of this class for each of the three strings input by the user and display the reversed string for each, in descending order of number of vowels in the string.
4. **Python for Data Science:** Load *Titanic Dataset* into one of the data structures (*NumPy* or *Pandas*). Perform data pre-processing on this dataset. Create dataframes, perform computations and visualize the results appropriately.
5. **Python File Handling & List Comprehension:** Write a python program to read contents of a file (filename as argument) and store number of occurrences of each word in a dictionary. Display the top 10 words with most number of occurrences in descending order. Store the length of each of these words in a list and display the list. Write a one-line reduce function to get the average length and one-line list comprehension to display squares of all odd numbers and display both.
6. **Python for Data Science:** Download the *Weather Dataset* from any online repository.
 - a) Change the Column Attributes in Data-frame to read as specified. E.g.

| Old Attribute | New Attribute | Old Attribute | New Attribute |
|---------------------------|---------------|---------------------|----------------|
| Mean Sea Level PressureIn | mean_pressure | Max VisibilityMiles | max_visibility |
 - b) Use the inbuilt functions to evaluate and analyze the weather data set.
 - c) Perform suitable data visualizations.
7. **JavaScript – Client Side Validation:** Design any two case studies that creates and validates a HTML form at the client side using Javascript. Case Study examples could be Grade Calculation from student marks entered, Bakery Menu & Price calculation of items bought etc.
8. **Introduction to JavaScript (JS):** Dynamically loading JSON data - Implement a HTML+JS application that has a JSON Array with details of different kinds of data. Example: Model, Name, Price, Year. Display details of each vehicle dynamically by only showing details of the vehicle that the user has selected (via mouse-over).
9. **Python and JavaScript - Form Validation:** Design any case study that creates and validates a HTML form. On the server side, use Python to display the details entered by the user at the client side along with appropriate validation messages. Use Display appropriate messages if error occurs, and redirect to the another HTML page if successful.

- 10. Python and JavaScript - ATM Application:** Design a HTML form that displays user's current balance, an input field to enter amount and buttons to withdraw or deposit money. Validate the form such that
 - i) Negative amount cannot be entered and Users cannot withdraw more than 5000 at one time
 - ii) Users cannot withdraw amount greater than their balance and cannot deposit more than 10000 at one time. Also users can perform at most 5 transactions. Update the balance accordingly and ensure relevant data is not lost on closing the browser.
- 11. Python and JavaScript - Shopping Cart Application:** Design a simple Shopping Cart application which allows users to add items to their cart from a list of products. Allow users to view their cart (items and quantities of each). Ensure that items in the cart persist even after closing the application. On selecting buy, print out a bill of items in the cart. Perform any necessary validation. Demonstrate data persistence even after the browser is closed.
- 12. Python for Data Science - Data Science and Machine Learning on *Boston Housing dataset*** - Download Boston Housing dataset from online archives. Load the dataset into one of the data structures (*NumPy* or *Pandas*). Visualize the dataset (Features vs price plot) using *matplotlib*. Apply Linear Regression ([Source to learn:scikit](#)) on the loaded dataset to predict prices of new data.

Reference Books:

1. Paul Barry, Head First Python, O'Reilly Publication, 2010.
2. Shelley Powers, Learning JavaScript, O'Reilly Publication, 2nd Edition, 2012

Course Outcomes:

1. Use internal and external Python libraries, data structures, functions inherent to Python in-order to handle data and use JavaScript to develop command line applications that create child processes to execute other programs and capture their output.(PO-1, 2, 5, 6, 10,12) (PSO-1, 2, 3)
2. Apply Python as a scripting language to analyze huge datasets, apply data science related statistics on datasets (PO-1, 2, 5, 6, 10, 12) (PSO-1, 2, 3)
3. Design and develop a simple web application with client-side JavaScript, server-side Python, using Flask – a micro-framework and develop JavaScript applications that can manipulate HTML pages dynamically, in response to user driven events and do client-side form validations. (PO-1, 2, 5, 6, 10, 12) (PSO-1, 2, 3)

DIGITAL IMAGE PROCESSING

Course Code: ISEA1

Credits: 4:0:0:0

Course Coordinator: Dr. Megha. P. Arakeri

Contact Hours: 56L

UNIT-I

Introduction: Digital image processing, Imaging modalities, Fundamental steps in image processing, Components of image processing system, Applications of image processing; **Digital Image Fundamentals:** Image formation model, Sampling and quantization, Relationships between pixels, Mathematical tools used in image processing.

UNIT-II

Spatial Filtering: Intensity transformation functions, Histogram processing, Fundamentals of spatial filtering, Smoothing spatial filters, Sharpening spatial filters; **Filtering in the Frequency Domain:** 2-D DFT, FFT, Basics of filtering in frequency domain, Image smoothing and sharpening using frequency domain filters.

UNIT-III

Image Segmentation: Fundamentals, Detection of isolated points, line and basic edge, Thresholding, Region-based segmentation; **Representation and Description:** Representation (border following, chain codes, minimum-perimeter polygons) Boundary descriptors (simple descriptors, shape numbers, Fourier descriptors), Region descriptors (simple descriptors, topological descriptors, texture).

UNIT-IV

Object Recognition: Patterns and pattern classes, Decision-theoretic methods, Structural methods; **Image Compression:** Fundamentals, Compression methods (Huffman, Arithmetic, Run-length, LZW), Digital image watermarking.

UNIT-V

Color Image Processing: Color models, Full-color image processing, Color transformations, Smoothing and sharpening. Segmentation in HIS and RGB space; **Morphological Image Processing:** Erosion and Dilation, Opening and closing, Hit-or-miss transform, Morphological algorithms.

Text Book:

1. Rafael C.Gonzalez, Richard E. Woods , “Digital Image Processing”, 3rd ed., Pearson.

References:

1. Anil K. Jain, “Fundamentals of Digital Image Processing”, Pearson Education, 2001.
2. B. Chanda and D. Dutta Majumdar ,“Digital Image Processing and Analysis”, PHI, 2003.

Course Outcomes:

1. Describe the fundamental concepts of a digital image processing(PO-1) (PSO-1).
2. Apply spatial and frequency domain filters to improve the quality of the image (PO-1,2,3)(PSO-1,2).
3. Employ segmentation, description, compression, recognition, color models and morphological techniques in processing the image(PO-1,2,3)(PSO-1,2).
4. Analyze wide range of problems related to image processing and provide solutions through suitable algorithms (PO-1,2,3,5) (PSO-1,2).
5. Exhibit teamwork and communication skills through course projects(PO-9,10) (PSO- 3).

NATURAL LANGUAGE PROCESSING

Course Code: ISEA2

Course Coordinator: Rajeshwari S B

Credits: 4:0:0:0

Contact Hours: 56L

UNIT-I

Introduction: Knowledge in Speech and Language Processing, Ambiguity, Models and Algorithms; Language, Thought, and Understanding; The State of the Art and The Near-Term Future; **Regular Expressions and Automata; Morphology and Finite-State Transducers:** Lexicon-free FSTs: The Porter Stemmer, Human Morphological Processing.

UNIT-II

N-grams: Counting Words in Corpora, Smoothing, N-grams for Spelling and Pronunciation, Entropy; **Word Classes and Part-of-Speech Tagging:** Part-of-Speech Tagging, Rule-based Part-of-speech Tagging, Stochastic Part-of-speech Tagging, Transformation-Based Tagging;

UNIT-III

Context-Free Grammars for English: Constituency, Context-Free Rules and Trees, Sentence-Level Constructions. **Parsing with Context-Free Grammars:** The Earley Algorithm; **Features and Unification:** Feature Structures, Unification of Feature Structures, Features Structures in the Grammar, Implementing Unification, Parsing with Unification Constraints;

UNIT-IV

Lexicalized and Probabilistic Parsing: Probabilistic Context-Free Grammars, Problems with PCFGs. **Representing Meaning:** First Order Predicate Calculus, Some Linguistically Relevant Concepts, Related Representational Approaches, Alternative Approaches to Meaning;

UNIT-V

Semantic Analysis: Syntax-Driven Semantic Analysis; **Lexical Semantics:** Relations Among Lexemes and Their Senses, WordNet: A Database of Lexical Relations. **Discourse:** Reference Resolution, Text Coherence, Discourse Structure.

Text Book:

1. Daniel Jurafsky and James H Martin, "Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition", Prentice Hall, 2nd Edition, 2008.

References:

1. Steven Bird, Ewan Klein, Edward Loper, "Natural Language Processing with Python- Analyzing Text with the Natural Language Toolkit", O'reilly Publications , 2009
2. Tanveer Siddiqui, U.S. Tiwary, "Natural Language Processing and Information Retrieval", Oxford University Press, 2008.

Course Outcomes:

1. Illustrate how speech and language technology relies on formal models to capture Knowledge. (PO- 1, 2, 3) (PSO-1,2)
2. Explain the way N-gram and part-of-speech tagging is used for spelling and pronunciation processing. (PO- 2, 3, 5, 9, 10) (PSO-1, 2)
3. Depict unification and early algorithms to capture syntactic information. (PO- 1, 2, 3, 5, 9, 10) (PSO-1, 2)
4. Explore on representations used to bridge the gap between languages and commonsense Knowledge (semantic processing). (PO- 1, 2, 3, 5, 9, 10) (PSO-1, 2)
5. Emphasize on how probabilistic context free grammar and first order predicate calculus are used to represent meaning (PO- 2, 3, 5, 9, 10) (PSO-1,2)

EMBEDDED COMPUTING SYSTEMS

Course Code : ISEA3

Course Coordinator : George Philip

Credits : 4:0:0:0

Contact Hours : 56L

UNIT-I

Introduction to Embedded Systems:- Model of an Embedded System, Microprocessor vs. Microcontroller, Example: A simple temperature monitor, Classification of MCUs, Current Trends. **MCU**:- The processor, The Harvard Architecture, GPIO, Power on Reset, Brown Out Reset, Watch Dog Timer, Real Time Clock, Memory Types, Low Power Design.

UNIT-II

Elementary ideas of Sensors, ADCs, Actuators. **Buses and Protocols**:- Elementary ideas of Parallel, I2C, SPI, USB, IEEE 1394, RS-232, RS-422/RS-485, Ethernet, CAN, WLAN, ZigBee, Bluetooth.

UNIT-III

Program Design and Analysis:- Components for Embedded Programs, Models of programs, Compiler Optimizations, Program-level Performance Analysis, Software Performance Optimization, Program-level Energy and Power Analysis and Optimization, Analysis and Optimization of Program Size.

UNIT-IV

Processes and Operating Systems:- Multiple Tasks and Processes, Multirate Systems, Preemptive RTOS, **Priority-Based Scheduling** – Rate Monotonic Scheduling, Earliest Deadline First Scheduling, Evaluating OS Performance, Power Management and Optimization for Processes.

UNIT-V

Embedded Program Development, Integrated Development Environment, Compiler, Assembler, Builder, Disassembly, Linker, Simulator, Downloading the Hex file, Hardware Simulator. **Hardware Software Co-Design**, Embedded Product Development Lifecycle Management, Testing. **Embedded System Examples**:- Mobile Phone, Automotive Electronics, RFID, WISENET, Robotics, Biomedical Applications, BMI.

Text Books:

1. Lyla B Das; Embedded Systems: An Integrated Approach, Pearson Education, 2013.
2. Marilyn Wolf; Computers as Components: Principles of Embedded Computing System Design, 3e, Morgan Kaufmann, 2012.

References:

1. Peter Barry, Patrick Crowley; Modern Embedded Computing, 1e, Morgan Kaufmann, 2012.
2. Lori M. Matassa, Max Domeika; Break Away with Intel Atom Processors – A Guide to Architecture Migration, 1e, Intel Press, 2010.

Course Outcomes:

1. Illustrate the model of an Embedded System and identify the components that make up a modern Embedded Computing System. (PO-1) (PSO-1)
2. Interpret the function of Sensors, ADCs, and Actuators; identify buses and protocols for specific applications. (PO-1) (PSO-1)
3. Analyze embedded system programs for performance issues; identify and apply optimization techniques. (PO-1,2,7) (PSO-1,2)
4. Identify real-time embedded system requirements and its implications in the design of RTOS and power management. (PO-1,2,3,4,5,7) (PSO-1,2)
5. Discuss the Embedded System Development process and identify example embedded systems. (PO-1,5) (PSO-1,2)

SYSTEM SOFTWARE

Course Code: IS61

Course Coordinator: Pushpalatha M N

Credits: 4:0:0:0

Contact Hours: 56L

UNIT-I

SIC,SIC/XE Architecture and Assembler: Introduction, System Software and Machine Architecture, Simplified Instructional Computer (SIC) - SIC Machine Architecture, SIC/XE, Basic Assembler Function, A Simple SIC Assembler, Assembler Algorithm and Data Structures, SIC object code generation.

UNIT-II

Assembler Features and Design Options: Machine Dependent Assembler Features - Instruction Formats & addressing Modes, Program Relocation, Machine Independent Assembler Features – Literals, Symbol-Definition Statements, Expression, Program Blocks, Control Sections and Programming Linking, SIC/XE object code generation. Assembler Design Operations - One-Pass Assembler, Multi-Pass Assembler, Implementation Examples – MASM Assembler

UNIT-III

Loader Functions, Features and Design Options: Basic Loader Functions - Design of an Absolute Loader, A Simple Bootstrap Loader, Machine-Dependent Loader Features – Relocation, Program Linking, Algorithm and Data Structures for a Linking Loader; Machine-Independent Loader Features - Automatic Library Search, Loader Options, Loader Design Options - Linkage Editor, Dynamic Linkage, Bootstrap Loaders, Implementation Examples - MS-DOS Linker.

UNIT-IV

Macro Processor Functions, Features and Design Options: Basic Macro Processor Functions - Macro Definitions and Expansion, Macro Processor Algorithm and Data Structures, Machine-Independent Macro Processor Features -Concatenation of Macro Parameters, Generation of Unique Labels, Conditional Macro Expansion, Keyword Macro Parameters, Macro Processor Design Options Recursive Macro Expansion, General-Purpose Macro Processors, Macro Processing Within Language Translators, Implementation Examples - MASM Macro Processor, ANSI C Macro Processor.

UNIT-V

LEX and YACC Tools: Lex and Yacc - The Simplest Lex Program, Recognizing Words With LEX, Grammars, Parser-Lexer Communication, A YACC Parser, The Rules Section, Running LEX and YACC, Using LEX – Regular Expression, Using YACC – Grammars, Recursive Rules Shift/Reduce Parsing, What YACC Cannot Parse, A YACC Parser - The Definition Section, The Rules Section, The LEXER, Compiling and Running a Simple Parser, Arithmetic Expressions and Ambiguity.

Text Books:

1. Leland.L.Beck, System Software, 3rd Edition, Addison-Wesley, 2002
2. John.R.Levine, Tony Mason and Doug Brown, Lex and Yacc, O'Reilly, SPD, 2004

Reference:

1. D.M.Dhamdhere, System Programming and Operating Systems, 2nd Edition, Tata McGraw Hill, 2000

Course Outcomes:

1. Describe the architecture of SIC, SIC/XE machine and to design the programs on those machine. (PO-1,2,3) (PSO-1,2)
2. Apply the assembler algorithm to generate the object code for SIC/XE. (PO-2,3) (PSO-1,2)
3. Explain the working of different types of Loaders. (PO-2, 3) (PSO-1,2)
4. Analyze the concepts of macro processing and its various features. (PO-2,3) (PSO-1,2)
5. Design the programs using LEX and YACC tools. (PO-1,2, 3, 5) (PSO-1,2)

OBJECT ORIENTED ANALYSIS AND DESIGN PATTERNS USING C++/JAVA

Course Code: IS62A/IS62B
Course Coordinator: Shashidhara H S

Credits: 3:0:0:1
Contact Hours: 42L

UNIT-I

The Object Oriented Paradigm – Functional Decomposition, The Problem with Requirements, Dealing with Changes, The OO Paradigm; **The UML—The Unified Modeling Language**: Overview, What Is the UML?, Why Use the UML?, The Class Diagram, Interaction Diagrams; **A Problem That Cries Out for Flexible Code**: Overview, Extracting Information from a CAD/CAM System, Understand the Vocabulary, Describe the Problem, The Essential Challenges and Approaches, Summary, A Standard Object-Oriented Solution: Overview, Solving with Special Cases; **Self-Study**: UML modeling for different problem scenarios; Understand the limitations of traditional Object Oriented Design

UNIT-II

An Introduction to Design Patterns: Overview, Design Patterns Arose from Architecture and Anthropology, Moving from Architectural to Software Design Patterns, Why Study Design Patterns?, Other Advantages to Studying Design Patterns, Summary; **The Facade Pattern**: Overview, Introducing the Façade Pattern, Learning the Façade Pattern, Field Notes: The Façade Pattern; **The Adapter Pattern**: Overview, Introducing the Adapter Pattern, Learning the Adapter Pattern, Field Notes: The Adapter Pattern; **Expanding Our Horizons – Objects**: Traditional Vs New Views, Encapsulation: Traditional Vs New Views, Finding Varying Concept and Encapsulating, Commonality and Variability Analysis and Abstraction; **Self-study**: Simple Factory

UNIT-III

The Strategy Pattern: Overview, The International E-Commerce System Case Study: Initial Requirements, Handling New Requirements, The Strategy Pattern; **The Bridge Pattern**: Overview, Introducing the Bridge Pattern, Learning the Bridge Pattern – An example, An Observation About Using Design Patterns, Learning the Bridge Pattern – Deriving It, The Bridge Pattern in retrospect; **The Abstract Factory Pattern**: Overview, Introducing the Abstract Factory Pattern, Learning the Abstract Factory Pattern – An example, Learning the Abstract Factory Pattern – Implementing It; **Self-study**: Command Pattern

UNIT-IV

How Do Experts Design – Building by Adding Distinctions; **The Principles and Strategies of Design Patterns**: The Open-Closed principle, Designing from Context, Encapsulating Variation, Abstract classes vs Interfaces, The principle of Healthy Skepticism; **Commonality And Variability Analysis: Application Design and Solving CAD/CAM Problem with CVA**; The Decorator Pattern: Overview, A Little More Detail, The Decorator Pattern, Applying the Decorator Pattern to the Case Study, Another Example: Input/Output; **The Observer Pattern**: Overview, Categories of Patterns, More Requirements for the International E-Commerce Case Study, The Observer Pattern, Applying the Observer to the Case Study; **The Template Method Pattern**: Overview, More Requirements for the International E-Commerce Case Study, The Template Method Pattern, Applying the Template Method to the International E-Commerce Case Study, Using the Template Method Pattern to Reduce Redundancy; **Self-study: Iterator Pattern**

UNIT-V

Lessons from Design Patterns: Factories – Factories, The Universal Context Revisited, Factories Follow our Guidelines; **The Singleton Pattern and the Double-Checked Locking Pattern**: Overview, Introducing the Singleton Pattern, Applying the Singleton Pattern to the Case Study; **A Variant**: The Double-Checked Locking Pattern, Reflections, Use the Singleton and Double-Checked Locking Patterns if applicable for a given problem; **The Factory Method Pattern**: Overview, More Requirements for the Case Study, The Factory Method Pattern, Factory Method Pattern and Object-Oriented Languages; **Self-study**: Object Pool Pattern

Ope
n
clos
e
princ
iple

Text Book:

1. Alan Shalloway, James R Trot, “Design Patterns Explained – A New Perspective on Object-Oriented Design”, Pearson, 2nd Edition, 4th Impression 2010.

References:

1. Eric Freeman, Elisabeth Freeman, “Head First Design Patterns”, O’reilly Publications, October 2004, 1st Edition
2. Satzinger, Jackson, Burd, “Object Oriented Analysis and Design with Unified Process”, Thomson Learning, 1st Indian Reprint 2007.

Course Outcomes:

1. Design solution to given problem definition using UML notations (PO-3, 5, 10, 12) (PSO-1,2,3)
2. Apply Façade, Adapter patterns to given problem with traditional design (PO-2, 3, 4, 5, 10) (PSO-1,2,3)
3. Analyze why and when we require Strategy, Bridge, Abstract Factory and Command Patterns (PO-2, 3, 4, 5, 10, 12) (PSO-1,2,3)
4. Apply design pattern principles and strategies to find flexible design solution to a given problem (PO-2, 3, 4, 5, 10) (PSO-1,2,3)
5. Demonstrate creation of similar objects using factory pattern (PO- 3, 4, 5, 10) (PSO-1,2,3)

JAVA & J2EE

Course Code: IS63

Course Coordinator: Prashanth Kambli

Credits: 3:0:0:1
Contact Hours: 42L

UNIT-I

java.util Part 1: The Collections Framework: Collections Overview, The Collection Interfaces, The Collection Classes: The ArrayList Class, LinkedList Class, HashSet Class. Accessing a Collection via Iterator, Storing User-Defined Classes in Collections; **Self-study:** Working with Maps, Arrays, Why Generic Collections?

UNIT-II

SWING: The Java Foundation Classes, JFC Technologies, Swing features, Swing Components, Working with SWING, SWING basic containers, Buttons, Labels, Text Fields, Text Areas, Check Boxes, JApplet, Combo Box, Event Handling. Using Dialogs, Timers and sliders; **Self-study:** Tables, Progress Bars.

UNIT-III

Introduction to Servlet: What is a servlet? Why servlets? What can servlets do? Servlet v/s CGI, Architecture of Servlet. **Basics of Servlet application programming Interface:** Servlet Application Programming Interface. The Servlet Architecture, The Servlet Life Cycle. **Working with Databases:** JDBC, Advantages of JDBC, JDBC Architecture, Types of JDBC Drivers, how to access database, Using Databases, Connecting to Database. **Java Threads:** What is multi-threading? The Thread Control methods, The thread life cycle, The main Thread, Creating the thread; **Self-study:** A simple hello world Servlet, Performing Database Operations, Extending the thread class.

UNIT-IV

Introduction to Java Server Pages: JSP v/s Servlets, Life cycle of a JSP Page. **Getting started with JSP:** JSP Elements, A Sample JSP Page. **Java Server Pages Document:** Directives, Scripting Elements, A Sample JSP Document. **Action elements of JSP:** Including other files, forwarding JSP page to another page, passing parameters for other actions. **Introduction to Enterprise JavaBeans:** Enterprise Bean architecture, features and benefits, types of enterprise beans, writing enterprise bean; **Self-study:** Beginning with enterprise java beans.

UNIT-V

Networking: Networking Basics, The Networking Classes and Interfaces, InetAddress, Factory Methods, Instance Methods, Inet4Address and Inet6Address, TCP/IP Client Sockets, URL, URLConnection, HttpURLConnection, The URI Class, Cookies, TCP/IP Server Sockets, Datagrams; **Self-study:** Datagram Socket, Datagram Packet, A Datagram Example

Text Books:

1. Herbert Schildt, „The Complete Reference Java (J2SE 5 Edition)“, TATA McGRAW-HILL Edition 2005.
2. Ivan Bayross, Sharanam Shah, Cyntiha Bayross and Vishali Shah, „Java EE 5 for Beginners“, SPD (Sharoff Publishers & Distributors Pvt. Ltd.), 2nd edition August 2008.

References:

1. Jim Keogh, „The Complete Reference J2EE“, TATA McGRAW-HILL Edition 2002.
2. B V Kumar, S Sangeetha, S V Subrahmany, J2EE Architecture, TATA McGRAW-HILL Edition 2007.

Course Outcomes:

1. Utilize the concepts of Collections Framework and interfaces of the Collections Framework to reduced development effort. (POs- 1,2,3) (PSO-1,2)
2. Apply event handling in Swing programs and create GUI interfaces using Java Applets(POs-1,2,3) (PSO-2)
3. Recognize the lifecycle of Servlets, Servlet Application Programming Interface and demonstrate working with JDBC (PO- 1,3,5) (PSO-1,2)
4. Develop a GUI Interface using Java server page and to learn Enterprise java Bean architecture (PO-1,3,5) (PSO-2).
5. Understand the concept of Networking in Java and demonstrate the working of Sockets and Packets using java.net library (POs- 1,2,3) (PSO-1,2)

MINI-PROJECT

Course Code: ISL64

Course Coordinator: Internal Guide

Credits: 0:0:6:0

Contact Hours: 64L

Guidelines:

At the end of the course, students are required to submit a mini-project from the domain of their choice.

An Internal Guide is allotted per student who guides and monitors the project progress.

The Internal Guide can arrange for doubt clarification classes if requested by his/her project student and records the same.

Industry Interaction courses can be introduced as part of Project Based Learning.

Course Outcomes (2 or more) are to be written per project and should map to following Program Outcomes and Program Specific Outcomes. Internal Guide can include other POs apart from the ones mentioned below if those POs are deemed suitable by them.

POs - 1, 2, 3, 11

(PO 5 - if a new tool is involved,

PO 6 - if issues involving/related to society, health, safety, legal and culture are addressed)

PSO - 1, 2, 3

In the case of Industry Interaction courses, Course Outcomes can be written per course and still be mapped to those POs and PSOs mentioned above plus any other POs as decided by the course coordinator. A mini project submission at the end of the course is mandatory.

Assessment:

CIE - Minimum 2 reviews of the project + any other relevant components (for 50 Marks)

SEE - Project Demonstration + any other component as decided by Internal Guide (for 50 Marks)

SYSTEM SOFTWARE LABORATORY

Course Code: ISL65

Course Coordinator: Myna A N

Credits: 0:0:1:0

Contact Hours:14P

PART-A

Implement the following using C/C++ for the SIC Machine:

1. Pass-1 of two-pass assembler.
2. Pass-2 of two-pass assembler to produce the object code.
3. Generate object program using the object code.
4. Absolute loader.
5. Relocating loader using bit mask.
6. One-pass macroprocessor.

PART-B

LEX Exercises:

1. Develop and implement a lex program to count the number of characters, words, spaces and lines in a given input file.
2. Develop and implement a lex program to count the number of identifiers, operators and comment lines in a given C program.
3. Develop and implement a lex program to recognize a valid arithmetic expression and identify the identifiers and operators present.
4. Develop and implement a lex program to count the number of vowels, consonants, numbers, special characters and lines in a given input file.
5. Develop and implement a lex program to count the number of
a) + ve and – ve numbers b) +ve and –ve fractions.

YACC Exercises:

1. Develop and implement a YACC program to recognize a valid arithmetic expression that uses operators +, -, *, and /.
2. Develop and implement a YACC program to recognize a valid variable, which starts with a letter, followed by any number of letters or digits.
3. Develop and implement a YACC program to evaluate an arithmetic expression involving operators +, -, *, and /.
4. Develop and implement a YACC program to recognize the strings of the form $a^m b^n$, where $m \geq 0, n \geq 0$.
5. Develop and implement a YACC program to accept an integer number with more than 3 digits.

Text Books:

1. Leland.L.Beck, System Software, 3rd Edition, Addison-Wesley, 1997.
2. John.R.Levine, Tony Mason and Doug Brown, Lex and Yacc, O'Reilly, SPD, 1999.
3. Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman, Compilers- Principles, Techniques and Tools, Second Edition, Addison-Wesley, 2007.

Reference:

1. D.M.Dhamdhere, System Programming and Operating Systems, 2nd Edition, Tata McGraw Hill, 1999

Course Outcomes:

1. Design and implement two pass assemblers for SIC machine. (PO- 1,2,3) (PSO-1,2)
2. Design and implement loaders. (PO- 1,2,3) (PSO-1,2)
3. Design and implement microprocessors.(PO- 1,2,3) (PSO- 1,2)
4. Design and implement Lex programs. (PO-1,2,3,5) (PSO- 1,2)
5. Design and implement YACC programs (PO- 1,2,3,5) (PSO- 1,2)

OBJECT ORIENTED ANALYSIS AND DESIGN PATTERNS LABORATORY

Course code and Title : ISL66

Course Coordinator: Jagadeesh Sai D

Credits : 0:0:1:0

Contact Hours: 14P

PART A

Case-study to understand the limitation of traditional Object Oriented Design and appreciate need for Design Patterns. Use UML Notations to design.

You are a fresh analyst deputed to design the software for Decathlon Chain of Stores in Karnataka. You are informed about the Business Logic of Point of Sales criteria by Ms.Veronica Lodge, a dynamic business tycoon operating out of Decathlon Mumbai. She informs you that there are different types of Customers of Decathlon namely, Regular Customers, Senior Citizens and First Time Customers. Regular Customers are given a discount of 12%, Senior Citizens 10% and First Time Customers 15%. Apart from this, based on the sales-index of previous day, a Store-level discount is determined every day. This is dynamic. E.g.Rs.100 off for every purchase above Rs.2000. Using the Object Oriented Principles of Encapsulation, Abstraction, Inheritance, Composition and Aggregation that you have studied until this semester, give at least two ways to design this system.

PART B

Common Case Study for Q#1 to Q#9

'Decathlon' is a Sports retail-store started in France. Today it spreads across 22 countries & has 900 outlets in these countries. It has a 'Point of Sale' software system called 'Decathlon POS', which uses various kinds of 3rd-party software sourced locally from the various countries they are established. You are a software consultant for Decathlon, in Bangalore, with a team of consultants reporting to you. When you analyze your answer for choosing a pattern, explain wherever applicable, keeping in mind the following four design principles:

- Separation of concerns
- Program to an interface, not a concrete implementation
- Prefer composition over inheritance
- Open-Close principle (Open for extension, Closed for modification)

1. **Adaptor (Structural):** To establish the 1st Decathlon store in Mauritius, you go along with Mr. Satya Nadella, an expert in finding 3rd-party partners. For e.g. a 3rd-party Tax-Calculator system to cater to the specifics of Sales and VAT (Value-added services Tax) tax calculations in different countries. He finds a 3rd-party Tax-Calculator system called 'MauriTAX' in Port Louis. The problem is, the APIs used by 'MauriTAX' for tax-calculation is fixed & cannot be changed. *The 'MauriTAX' APIs are incompatible with 'Decathlon POS'.* How will you use the Adaptor Pattern to design & implement?
2. **Strategy (Behavioural):** How will you use the Strategy Pattern to tackle the limitations of traditional Object Oriented Design highlighted in PART A? *The design must handle varying price-schemes having different pricing algorithms.* Design & implement.
3. **Factory Method (Creational):** The 'Decathlon POS' software system classifies its customers as senior-citizens (60 and above), First-Time customers, Regular Customers. There is a very high possibility that the Customer Type hierarchy will vary, depending upon the sales-pattern. **For e.g.** there could be the need to introduce new categories based on the customer gender, different age groups for kids (0-5, 6-12), teenagers (13-19) and age groups between 20 to 60(Twenties, 30s, 40s and 50s). You are advised by Mr.Sundar Pichai, the technical architect of your team, whom you trust, to use Factory Method Pattern in order to instantiate the above Customer Type hierarchy of concrete implementation of objects. Design and implement using this.
4. **Bridge (Structural):** You get a call from Ms.Masaba Gupta of Bangalore Decathlon office that there is a policy decision made globally to introduce discount slabs for a whole month twice in a year. The discount month will be in January and July after reviewing the sales made from Feb to June (first five months) and Aug to December (last five months) respectively. It is decided to provide four slabs of discounts in 2017, namely, 30%, 25%, 20% and 15%, based on the sports item purchased. **For e.g.** all tennis rackets could have a 20% discount while cricket bats could only have a 15% discount. All exercise tread-mills could be given a 30% discount while boxing-gloves could have a 25% discount. Point to be noted here is that, the slabs of discount may not remain the same in 2018. It is likely to vary year after year. The 'Decathlon POS' software system classifies its customers as Senior-Citizens (60 and above), First-Time Customers, Regular Customers as of now. There is a very high possibility that the Customer Type hierarchy will vary, depending upon the sales-pattern. **For e.g.** there could be the need to introduce new categories based on the customer gender.

Use the Bridge Pattern to design & implement, *so that both the Customer Type hierarchy of classes as well as the Discount Percentage hierarchy of classes can both vary independently?* That is, they

are not tied to each other.

- 5. Observer (Behavioural):** There will be different discounts being offered for the sports items in Decathlon Stores across the globe for different festivals being celebrated in the various countries these stores are established. Assume that the Decathlon Chain of Stores fixes a particular discount slab for its items for a festival of a country.
Use the Observer Pattern to design and implement a system to notify the customers of the Decathlon stores of that country about the various festival / seasonal discount rates as and when they are announced.
- 6. Façade (Structural):** You get a call from Ms.Betty Cooper of Bangalore Decathlon office that there is a policy decision made globally to incorporate some new rules for ‘Process Sale Use-Case’. **For e.g.** if payment is made via gift-certificate, the customer can buy only one item for the amount in the certificate. No other items can be bought with that gift-certificate. There must be no cash-back to the customer if the item costs less than amount specified in the gift-certificate. If the item costs more, the excess payment can be accepted via cash only & not credit / debit cards. When a new sale is created, these rules must become effective. You come to know from Mr. Satya Nadella, an expert in finding 3rd-party partners, that the Italian Competitor for Decathlon called ‘Sport 2000’ has a ready-made ‘rule-engine’ sub-system for this, whose specific implementation details is not known yet, as the business heads of Decathlon & Sport 2000 are chalking out the software purchase terms. This Sport 2000 rule-engine will be responsible for evaluating a set of rules against an operation & indicating if any of the rules invalidated the operation (e.g. ‘makeNewSale’ operation).
How will you use the Façade pattern to provide a common unified interface to a dissimilar set of implementations, developed by a 3rd-party vendor, the implementation details are not known to you?
- 7. Abstract Factory (Creational):** As an analyst in charge of designing the Decathlon POS Software, you realize the need to streamline the creation of objects belonging to different products in the Decathlon store. There are two major categories of products:
- a) For differently abled sports enthusiasts
 - b) For able-bodied sports enthusiasts
- In each of the above categories there are products for outdoor adventure sports (e.g. trekking, para-gliding, bungee-jumping etc.), outdoor regular games (cricket, football, baseball etc.) indoor regular games (table tennis, squash etc.). There is a possibility of further class/object instantiation explosion with categories such as male & female sports enthusiasts and different equipment for them. *Objects need to be instantiated based on these categories.* Design & implement using Abstract Factory.
- 8. Decorator(Behavioural – Structural according to GoF):** There is an existing interface method in the Decathlon POS software system called ‘getCurrentStock’ which is implemented by two concrete classes ‘IndoorSports’ and ‘OutdoorSports’, to get the number of stocks for the sports items belonging to these respective categories. On studying the Decathlon POS system, you as an analyst realize the need to get sports stock update of various items within:
- IndoorSports - ‘GamesOnTable’ (e.g. Table Tennis, Billiards, Snooker etc.)
‘BoardGames’ (e.g. Carom, Chess etc.) ‘CourtGames’ (e.g. Basketball, Badminton, Kabaddi etc.)
- OutdoorSports – ‘AdventureGames’ (e.g. trekking, para-gliding, bungee-jumping etc.)
‘StadiumGames’ (e.g. cricket, football, baseball etc.) ‘Athletics’ (e.g. different distances for running, high jump etc.)
- Use the Decorator pattern, decorating the ‘getCurrentStock’ method to design and implement this scenario.
- 9. Template Method (Behavioural):** To keep up with the customer convenience of online ordering Decathlon Chain of stores decides to have two modes of order-processing, namely ‘online’ and ‘offline’. Both modes have the same processing steps for order-processing, namely ‘selectItem’, ‘doPayment’ and ‘doDelivery’. But, the way these steps are done varies between the two modes.
selectItem – online – gives tabular depiction of price comparison of the item chosen. Offline – allows trying out of the items in the store
doPayment – online – net-banking payment; offline – pays through cash / swipe-card doDelivery – online – needs to pay the charges for shipping & delivery address; offline – collect at the counter.
Show how you as the analyst will use the Template Method pattern to design and implement this.
- 10. Singleton (Creational):** A Browser’s history has data of all the visited URLs across all tabs and windows of a browser. The history is saved such that the data persists even after closing the browser. How would you use Singleton Pattern to implement Browser History such that on visiting a URL on any open tab of a browser the URL gets added to the existing history?

Text Book:

1. Alan Shalloway, James R Trot, "Design Patterns Explained – A New Perspective on Object-Oriented Design", Pearson, 2nd Edition, 4th Impression 2010.

References:

1. Eric Freeman, Elisabeth Freeman, "Head First Design Patterns", O'reilly Publications, October 2004, 1st Edition
2. Satzinger, Jackson, Burd, "Object Oriented Analysis and Design with Unified Process", Thomson Learning, 1st Indian Reprint 2007.

Course Outcomes:

1. Design the given problem using standard UML(PO- 2, 5) (PSO-1, 2, 3)
2. Identify maintenance issues in the existing solutions(PO-2, 6) (PSO -1, 2, 3)
3. Explain the use of design patterns(PO- 10) (PSO-1, 2, 3)
4. Apply Design patterns to the problems(PO- 2, 4, 11) (PSO-1, 2, 3)

JAVA & J2EE LABORATORY

Course Code: ISL67
Course Coordinator: Koushik S

Credits: 0:0:1:0
Contact Hours: 14P

1. Write a Program that simulates a telephone that records missed incoming calls. For each missed call, store the time of call, telephone number of origin, and name of the called if the name is available. For unlisted numbers, set the name to “private caller”. Choose or extend the most appropriate collection class and provide the following features.
 - a) Numbers are recalled in the order they arrive
 - b) Up to 10 numbers are recorded. When the eleventh call comes in, it is stored and the oldest call is deleted so that no more than 10 numbers are ever recorded.
 - c) After each number display, the user can select
 - a. to delete the call
 - b. to go on to the next missed call, or
 - c. to display the call details (number, caller name and time).
 - d) Delete the number if user specifies a number to delete.
 - e) Write a helper class to represent an incoming call with fields to hold the number, name of the caller, and time of the call. Write a tester call that stores the several numbers, simulate the user pressing the missed-calls button, and finally prints the entire collection of stored calls.
 2. Write a Java program using user-defined storage classes to create a book database and store it in a Collection List. Books collection should include title, author, publisher and price. Also write a method to sort the books in ascending order of price and store it in another List.
 3. Write a program that uses Java Swing and JDBC to create a stand-alone application:
 - a. Create two tables namely, Representative (RepNo, RepName, State, Comission, Rate) and Customer (CustNo, CustName, State, Credit_Limit, RepNo) in MySQL database.
 - b. Use appropriate Swing components to insert values in a form.
 - c. Use another form to display Representative's information whose Credit Limit is above 15,000.
 4. Write a Java Swing program to simulate a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with “stop” or “ready” or “go” should appear above the buttons in a selected color.
 5. Write a Java program to create a progress bar GUI which displays the copy status of writing contents of one file to another file. Terminate the progress bar on click of STOP button and display the contents of the file.
 6. Create a Servlet to file IT returns that accepts personal information, salary information and Tax deduction details from the user and write the information into a file. Also accept the name of the person and display in on the page.
 7. Write a JSP and Servlet Program to do the following to buy a T-Shirt online:
 - a. A set of checkboxes to select your T-Shirt accessories such as ‘belt’, ‘cap’, ‘hair-band’ etc.
 - b. A text area / text field to enter your T-Shirt tag-line
 - c. A Radio-button that allows the user to choose between T-Shirt with chest pocket and without.
 - d. A Combo Box to choose your T-Shirt color
 - e. Appropriate labels for these GUI Components
 - f. A Button called “Click Me” which when pressed will
 - g. Insert the details entered into a table called ‘TShirts’.
 - h. An OrderNo is generated by adding ‘1’ to the existing ‘OrderNo’
 - i. If ‘TShirts’ table is empty the initial value of ‘OrderNo’ is 100.
 - j. This ‘OrderNo’ is also inserted into the ‘TShirts’ table
 - k. Display all the records of the ‘TShirts’ table in tabular form
- PS: Frontend display should be in JSP and the business logic should be written in Servlet Class.
8. Create a Telephone Directory Application using Servlet that searches the database based on phone number or name. Also show database table creation with inserting 2-3 values to the table.
 - a. Database Name: OnlineDirectory
 - b. Table Design:
 - i. Table Name: Telephone_Directory
 - ii. Attributes: Phone_Number, Name, Address, Company, Pin_Code.
 9. Write a Java Program that creates two **threads** object of Thread class. Where one thread asks the user to enter a number not less than four digits. Split the digits of the number and display in **words** the value of the number. Ex: 1 – One. Second thread finding the number of vowels in a word. Ex: JAVA – Vowel - A, Count – 2.
 10. Write a program using **JSP** that helps a student to calculate the income tax on various annual incomes that he will be earning when he gets a job.
 - a. *Login.html* will call *dataCapture.jsp* that should do the following:

- i. Use Java Collections to make a list of valid users and facilitate user login functionality.
 - ii. Give a personalized Welcome message and display today's date.
 - iii. Have a Text Entry with label 'Name' to enter the name of the user.
 - iv. Give a List of Organizations to choose 'Place of Work'
 - v. Provide a Male or Female option to choose the 'Gender'
 - vi. Have a Text Entry with label 'Annual Income'
 - vii. Give a Submit button reading 'Calculate Tax'
- b. *CalculateTax.jsp* must calculate the interest based on the following business rules:
- i. Salary below 1,00,000 shall no have income-tax.
 - ii. Calculate 15% of tax on 1,00,001 – 5,00,000.
 - iii. Calculate 20% on 5,00,001 onwards.

PS: The final income tax along with the details of how it is calculated must be put in a session object and displayed to the user in *dataCapture.jsp*. All the income taxes calculated so far by the user, must be taken out of the session object and displayed, each time in *dataCapture.jsp* which has a link called '*Logout*' that destroys the session.

11. a. Create two tables Flight (Flight_Number, Airline_Name, Weekdays) and SeatReservation(Flight_Number, Date, Seat_Number, Customer_Name, Customer_Phone) in MySQL database.
 - b. Create JSP page *ReserveOnline.jsp* to reserve an airline seat and insert the values into the table *SeatReservation*. OnClick of Submit in *ViewDetails.jsp* display information about reservation. Validate the Flight_Number from already existing Flight database and generate random number for Seat_Number within the range 1-500.
 - c. Also create a link to display information of all the flights running on a particular day.
12. Write a program using java.net to transfer file from client socket to server socket.

Text Books:

1. Herbert Schildt, 'The Complete Reference Java (J2SE 5 Edition)', TATA McGRAW-HILL Edition 2005.
2. Ivan Bayross, Sharanam Shah, Cyntiha Bayross and Vishali Shah, 'Java EE 5 for Beginners', SPD (Sharoff Publishers & Distributors Pvt. Ltd.), 2nd edition August 2008.

References:

1. Jim Keogh, „The Complete Reference J2EE“, TATA McGRAW-HILL Edition 2002.
2. B V Kumar, S Sangeetha, S V Subrahmanya, J2EE Architecture, TATA McGRAW-HILL Edition 2007.

Course Outcomes:

1. Design and Develop Java programs using the Java concepts such as packages, inheritance, interfaces, exception handling, collection frameworks, Swings, Servlets, JSP, JDBC and Multithreading. (PO - 2, 4) (PSO – 1)
2. Apply java concepts in various domains. (PO - 4) (PSO – 2)
3. Produce a substantial written documentation. (PO - 10) (PSO - 3)

MACHINE LEARNING

Course Code: ISEB1

Course Coordinator: Dr. Mydhili K Nair

Prerequisites : Python, Probability Theory and Statistics

Credits: 4:0:0:0

Contact Hours: 56L

UNIT-I

Introduction to Machine Learning - What is Machine Learning? Examples of Machine Learning Applications.

Types of Machine Learning; Supervised Learning; The Machine Learning Process, A Note on Programming

Machine Learning Preliminaries - Terminology: Weight Space, The Curse of Dimensionality; Testing Machine Learning Algorithms; Turning Data into Probabilities: Minimizing Risk, The Naive Bayes Classifier; Basic Statistics: Averages, Variance and Covariance, The Gaussian; Bias-Variance Tradeoff

UNIT-II

Neural Networks - The Perceptron, Linear Separability, Linear Regression. **Multi-layer Perceptron** - Going Forwards, Going Backwards - Back Propagation of Error, Multi-layer perceptron in practice; Examples of using MLP: A Regression Problem, Classification with MLP, A Classification Example - the IRIS dataset, Time Series Prediction, Data Compression - Auto Associative Network; A recipe for using MLP; Deriving Back-propagation

UNIT-III

Dimensionality Reduction - LDA (Linear Discriminant Analysis); PCA (Principal Component Analysis); Factor Analysis; Independent Components Analysis; Locally Linear Embedding; ISOMAP. **Supervised Learning** - Learning a Class from Examples, VC (Vapnik-Chervonenkis) Dimension, PAC (Probably Approximately Correct) Learning, Noise, Learning Multiple Classes. **Bayesian Decision Theory** - Classification, Losses and Risks, Discriminant Functions, Utility Theory, Association Rules

UNIT-IV

Learning with Trees - Using Decision Trees, Constructing Decision Trees; CART - Classification and Regression Trees, Classification Example. **Probabilistic Learning** - Gaussian Mixture Models; Nearest Neighbour Methods. **Support Vector Machines** - Optimal Separation; Kernels; The SVM Algorithm; Extensions of the SVM

UNIT-V

Unsupervised Learning - The K-Means Algorithm; Vector Quantisation; The Self-Organizing Feature Map

Clustering - Introduction, Mixture Densities, K-Means Clustering, Expectation-Maximization Algorithm, Supervised Learning after clustering, Hierarchical Clustering, Choosing the Number of Clusters. **Reinforcement Learning** - Overview; Example - Getting Lost; Markov Decision Process; Values; Using Reinforcement Learning; Uses of Reinforcement Learning.

Text Books:

1. Stephen Marsland, "Machine Learning - An Algorithmic Perspective", Second Edition, CRC Press - Taylor and Francis Group, 2015
2. Ethem Alpaydin, "Introduction to Machine Learning", Second Edition, MIT Press, Prentice Hall of India (PHI) Learning Pvt. Ltd. 2010

References:

1. Christopher Bishop, "Pattern Recognition and Machine Learning", CBS Publishers & Distributors, 2010.
2. Mehryar Mohri, Afshin R, Ameet Talwalkar, "Foundations of Machine Learning", MIT Press, 2012.
3. Charu C. Aggarwal, "Data Classification Algorithms and Applications" , CRC Press, 2014
4. Charu C. Aggarwal, "Data Clustering Algorithms and Applications", CRC Press, 2014.
5. Kevin P. Murphy "Machine Learning: A Probabilistic Perspective", The MIT Press, 2012

Course Outcomes:

1. Recognize the preliminary concepts associated with machine learning such as its need, types, statistical terms and testing ML algorithms (PO1, PO9, PO10, PO12) (PSO1, PSO2, PSO3)
2. Employ neural networks in various scenarios by figuring out how multi-layer perception works (PO1, PO2, PO3, PO5, PO7, PO9, PO10, PO12) (PSO1, PSO2, PSO3)
3. Handle the nuances of supervised learning including Bayesian Theory and Decision Trees, appreciate the need for dimensionality reduction and use the same.(PO1, PO2, PO3, PO5, PO7, PO9, PO10, PO12) (PSO1, PSO2, PSO3)
4. Apply the concepts of CART - Classification & Regression Trees, probabilistic learning and SVM. (PO1, PO2, PO3, PO5, PO7, PO9, PO10, PO12) (PSO1, PSO2, PSO3)
5. Identify the need of unsupervised and reinforcement learning and apply the same. (PO1, PO2, PO3, PO5, PO7, PO9, PO10, PO12) (PSO1, PSO2, PSO3)

COMPILER DESIGN

Course Code: ISEB2

Course Coordinator: Rajeshwari S B

Credits: 4:0:0:0

Contact Hours: 56L

UNIT-I

Introduction to compilers: Introduction, Theory of computer Languages, Design of a Language, Evolution of Compilers, stages of Compilation. **Lexical analysis:** Introduction, Alphabets and Tokens in Computer Languages, Representation of Tokens and Regular Expression, Token Recognition and Finite State Automata, Implementation, Error Recovery.

UNIT-II

Syntax Analysis-I: Introduction, Context-free Grammar and Structure of Language, Parser and its Types, Top-down Parsers

UNIT-III

Syntax Analysis-II: Bottom up-Parsers, Implementation, Parser Generator Tool(Yacc), Error Handling

UNIT-IV

Intermediate Code generation: Introduction, Need for Intermediate Code, Types of Intermediate Code, Representations of All Language Constructs by Three-address code, Grammar Symbols and Attributes, Semantic Analysis, Semantic Routines for Intermediate Code Generation. **Optimization:** Introduction, Hints on Writing optimized Code at User Level, Construction of Basic Blocks and Processing, Data-flow Equations for Blocks with Backward Flow Control, Principal Sources of Optimization and Transformations, Alias, Procedural Optimization, Loops in Flow Graphs, Loop Optimization

UNIT-V

Code generation: Introduction, Issues in Code Generation, Target Machine Architecture, Subsequent Use Information, Simple Code Generator, Register Allocation, Directed Acyclic Graph Representation, Code Generation from Intermediate Code, Peephole Optimization. **Compiler Writing tools:** Introduction, Lexical Tools, Syntactic Tools.

Text Books:

1. Alfred V. Aho, Ravi Sethi and Jeffrey D. Ullman, “Compilers principles, Techniques and Tools” Pearson Education. 2013
2. Muneeswaran, “Compiler Design”, Oxford University Press, 2013

References:

1. K.D. Cooper, and Linda Torczon, Engineering a Compiler, Morgan Kaufmann, 2004.
2. K.C. Louden, Compiler Construction: Principles and Practice, Cengage Learning, 1997.

Course Outcomes:

1. Depict representation of tokens and token recognition in lexical analysis phase.(PO-1,2,3,5,9,10) (PSO-1,2)
2. Analyze the sentence of a language for syntax checking using top-down parsing technique. (PO-2,3,5,9,10) (PSO-1,2)
3. Describe parser generator tool, various bottom-up parsing and error-handling techniques. (PO-2,3,5,9,10) (PSO-1,2)
4. Explain the process involved in generation of intermediate code and code optimization techniques. (PO-2,3,5,9,10) (PSO-1,2)
5. Illustrate generation of machine code from intermediate code and the functionality of compiler writing tools. (PO-2,3,5,9,10) (PSO-1,2)

DISTRIBUTED STORAGE TECHNOLOGIES

Course Code: ISEB3

Course Coordinator: Dr. Siddesh G M

Credits: 4:0:0:0

Contact Hours: 56L

UNIT-I

Introduction: Information Storage, Evolution of Storage Architecture, Data Centre Infrastructure, Virtualization and Cloud Computing. Data Centre Environment: Application, DBMS, Host, Connectivity, Storage, Disk Drive Components, Disk Drive Performance, Host Access to Data, Direct-Attached Storage, Storage Design Based on Application, Disk Native Command Queuing, Introduction to Flash Drives.

UNIT-II

Data Protection: RAID Implementation Methods, Array Components, Techniques, Levels, Impact on Disk Performance, Comparison, Hot Spares. Intelligent Storage System: Components, Storage Provisioning, Types.

UNIT-III

Fibre Channel Storage Area Networks: FC Overview, Evolution, Components, FC Connectivity, Ports, FC Architecture, Fabric Services, Login Types, Zoning, FC Topologies, Virtualization in SAN. IP SAN and FCoE: iSCSI, FCIP, FCoE.

UNIT-IV

Network-Attached Storage: Benefits, Components, NAS I/O Operation, Implementations, File Sharing Protocols, Factors Affecting NAS Performance, File-Level Virtualization. Object Based and Unified Storage: Object Based Storage Devices, Content Addressed Storage, CAS Use Cases, Unified Storage.

UNIT-V

Business Continuity: Information Availability, Terminology, Planning Lifecycle, Failure Analysis, Impact Analysis, Solutions. Cloud Computing: Cloud Enabling Technologies, Characteristics, Benefits, Service Models, Deployment Models, Infrastructure, Challenges, Adoption Considerations. Securing the Storage Infrastructure: Framework, Risk Triad, Domains. Managing the Storage Infrastructure: Monitoring, Management Activities, Management Challenges, Information Lifecycle Management, StorageTiering.

Text Book:

Somasundaram G, Alok Shrivastava, (EMC Education Services), Information Storage and Management, 2e, Wiley India, 2012, ISBN 9788126537501.

Reference:

Robert Spalding; Storage Networks: The Complete Reference, Tata McGraw Hill, 2003.

Course Outcomes:

1. Explain storage architectures, and the logical and physical components of storage infrastructure including storage subsystems (PO-1,2,5,9,10,12)(PSO-3)
2. Describe RAID and intelligent storage systems(PO-1,2)(PSO-3)
3. Illustrate storage networking technologies such as FC SAN, IP SAN, and FCoE (PO-1)(PSO-2)
4. Discuss NAS, and object-based and unified storage(PO-1)(PSO-3)
5. Explain business continuity, cloud computing, storage security, storage monitoring and management activities (PO-1,3,6,7)(PSO-3)