RAMAIAH
Institute of Technology

CURRICULUM

Academic Year 2021– 2022

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:

Dr. M. S. Ramaiah a philanthropist, founded 'Gokula Education Foundation' in 1962 with an objective of serving the society. M S Ramaiah Institute of Technology (MSRIT) was established under the aegis of this foundation in the same year, creating a landmark in technical education in India. MSRIT offers 13 UG programs and 15 PG programs. All these programs are approved by AICTE. All the UG programs & 09 PG programs are accredited by National Board of Accreditation (NBA). The institute is accredited with 'A' grade by NAAC in 2014. University Grants Commission (UGC) & Visvesvaraya Technological University (VTU) have conferred Autonomous Status to MSRIT for both UG and PG Programs till the year 2029. The institute is a participant to the Technical Education Quality Improvement Program (TEQIP), an initiative of the Government of India. The institute has 380 competent faculty out of which 60% are doctorates. Some of the distinguished features of MSRIT are: State of the art laboratories, individual computing facility to all faculty members, all research departments active with sponsored funded projects and more than 300 scholars pursuing Ph.D. To promote research culture, the institute has established Centre of Excellence for Imaging Technologies, Centre for Advanced Materials Technology & Schneider Centre of Excellence. **M S Ramaiah Institute of Technology has obtained "Scimago Institutions Rankings" All India Rank 65 & world ranking 578 for the year 2020.**

The Centre for Advanced Training and Continuing Education (CATCE), and Entrepreneurship Development Cell (EDC) have been set up on campus to incubate startups. **M S Ramaiah Institute of Technology secured All India Rank 8th for the year 2020 for Atal Ranking of Institutions on Innovation Achievements (ARIIA), an initiative of Ministry of Human Resource Development (MHRD), Govt. of India.** MSRIT 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 good collection of book volumes and subscription to International and National Journals. The Digital Library subscribes to online e-journals from Elsevier Science Direct, IEEE, Taylor & Francis, Springer Link, etc. MSRIT is a member of DELNET, CMTI and VTU E-Library Consortium. MSRIT has a modern auditorium and several hi-tech conference halls with video conferencing facilities. It has excellent hostel facilities for boys and girls. MSRIT Alumni have distinguished themselves by occupying high positions in India and abroad and are in touch with the institute through an active Alumni Association.

As per the National Institutional Ranking Framework, MHRD, Government of India, M S Ramaiah Institute of Technology has achieved 65th rank among 1143 top Engineering Institutions of India for the year 2021 and 1st rank among VTU affiliated Institutions in Karnataka.

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, 2015 and reaccredited in 2018 under Tier-1 till 2022. 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, EPICS, expert lectures, partial course deliveries by industry experts and student interns to enhance the skills in emerging areas to keep an inclusive and diverse academic environment. Department is regularly conducting seminars, conferences and workshops for students and academicians in the emerging areas of Information Technology. Introduced EPICS in senior projects. Some of the laboratories have also been set up in collaboration with industries such as Intel, Microsoft, Apple, SECO, Honeywell, EMC^2, NVIDIA, IBM, Green Sense Werks, Tech Machinery Labs, Sesovera Tech Pvt. Ltd., and Ramaiah Medical College (Emergency department). Also, an echo system is built to initiate start-ups at the department level along with the mentorship. All the above potential activities have led 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 MS 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

Department of Information Science and Engineering shall create high quality IT Engineering Professionals for the betterment of society by:

- Providing education through an ever improving curriculum and effective pedagogy techniques.

- Encouraging extra and co-curricular activities to develop their overall personality along with technical skills.

- Collaborating with industry and academia for strengthening research, innovation and entrepreneurship ecosystem.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

Graduate will be able to:

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

PEO2: Productively engage with society by practicing research or entrepreneurship.

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: Apply Mathematical models, programming paradigms and software development practices to solve real world problems

PSO2: Adopt computing and communication models for developing IT solutions.

PSO3: Acquire data engineering skills to develop intelligent systems in a multidisciplinary environment.

Semester wise Credit Breakdown for B E Degree Curriculum

Semester / Course Category	First	Second	Third	Fourth	Fifth	Sixth	Seventh	Eighth	Total Credits
Basic Sciences (BSC)	9	8	4	4					25
Engineering Sciences (ESC)	11	10							21
Humanities, Social Sciences and Management (HSMC)		2			3		3		8
Professional Courses – Core (PCC)			21	21	15	11	10		78
Professional Courses– Elective (PEC)					3	6	6	3	15
Other Open Elective Courses (OEC)					3	3			6
Project Work (PROJ), Internship (IN)						4	1	14	22
Total Credits	20	20	25	25	24	24	20	17	175

SCHEME OF TEACHING
V SEMESTER

Sl. No.	Course Code	Course Name	Category	Credits				Contact Hours
				L	T	P	Total	
1.	IS51	Computer Networks	PC-C	4	0	0	4	4
2.	IS52	Database Management Systems	PC-C	3	1	0	4	5
3.	IS53	Software Engineering	PC-C	3	0	0	3	3
4.	IS54	Intellectual Property Rights	HSS	3	0	0	3	3
5.	ISE55X	Professional electives – 1	PE	3	0	0	3	3
6.	ISOE0X	Open electives – 1	OE	3	0	0	3	3
7.	ISL56	Computer Networks Lab	PC-C	0	0	1	1	2
8.	ISL57	Database Management Systems Lab	PC-C	0	0	1	1	2
9.	ISL58	Scripting Languages Lab	PC-C	0	0	2	2	4
			Total	**19**	**1**	**4**	**24**	**29**

L – Lecture (one hour)   T - Tutorial (Two hours)   P - Practical (Two hours)

Professional Electives:

1	ISE551	Digital Image Processing
	ISE552	Natural Language Processing
	ISE553	Internet of Things
	ISE554	Develop in Swift Fundamentals

Open Electives – 1:

1	ISOE06	Software Project Management
	ISOE07	Object Oriented Programming with C++

Note:

AICTE Activity Points to be earned by students admitted to BE program (For more details refer to Chapter 6 AICTE, Activity Point Program, and Model Internship Guidelines):

Every regular student, who is admitted to the 4 year degree program, is required to earn 100 activity points in addition to the total credits earned for the program. Students entering 4 years Degree Program through lateral entry are required to earn 75 activity points in addition to the total credits earned for the program. The activity points earned by the student shall be reflected on the students 8th Semester grade card. The activities to earn the points can be spread over the duration of the course. However, minimum prescribed duration should be fulfilled. Activity Points (non-credit) have no effect on SGPA/CGPA and shall not be considered for vertical progression.

Incase student fail to earn the prescribed activity points, Eight semester Grade Card shall be issued only after earning the required activity Points. Students shall be eligible for the award of degree only after the release of the Eight Semester grade card.

SCHEME OF TEACHING
VI SEMESTER

Sl. No.	Course Code	Course Name	Category	Credits				Contact Hours
				L	T	P	Total	
1.	IS61	Machine Learning	PC-C	3	1	0	4	5
2.	IS62	Object Oriented Analysis and Design Patterns	PC-C	3	0	0	3	3
3.	ISE63X	Professional elective - 2	PE	3	0	0	3	3
4.	ISE64X	Professional elective - 3	PE	3	0	0	3	3
5.	ISOE0X	Open elective - 2	OE	3	0	0	3	3
6.	IS65	Mini Project/ NPTEL	PC-C	0	0	4	4	-
7.	ISL66	Machine Learning Lab	PC-C	0	0	1	1	2
8.	ISL67	Object Oriented Analysis and Design Patterns Lab	PC-C	0	0	1	1	2
9.	ISL68	Java and J2EE Lab	PC-C	0	1	1	2	4
			Total	15	2	7	24	25

L – Lecture (one hour)   T - Tutorial (Two hours)   P - Practical (Two hours)

Professional Electives:

2	ISE631	Develop in Swift Data Collections
	ISE632	System Software
	ISE633	System Simulation and Modelling
	ISE634	Blockchain Essentials & Dapps

3	ISE641	Cloud Computing
	ISE642	Mobile Computing
	ISE643	Software Testing
	ISE644	Artificial Intelligence

Open Electives – 2:

1	ISOE08	Object Oriented Programming with Java
	ISOE09	Software Testing
	ISOE10	Data Science with Python

Note:

AICTE Activity Points to be earned by students admitted to BE program (For more details refer to Chapter 6, AICTE, Activity Point Program, and Model Internship Guidelines):

Every regular student, who is admitted to the 4 year degree program, is required to earn 100 activity points in addition to the total credits earned for the program. Students entering 4 years Degree Program through lateral entry are required to earn 75 activity points in addition to the total credits earned for the program. The activity points earned by the student shall be reflected on the students 8th Semester grade card. The activities to earn the points can be spread over the duration of the course. However, minimum prescribed duration should be fulfilled. Activity Points (non-credit) have no effect on SGPA/CGPA and shall not be considered for vertical progression.

Incase student fail to earn the prescribed activity points, eighth semester Grade Card shall be issued only after earning the required activity Points. Students shall be eligible for the award of degree only after the release of the Eight Semester grade card.

COMPUTER NETWORKS

Course Code: IS51   **Credit: 4:0:0**
Prerequisite: Data Communications   **Contact Hours: 56L**
Course Coordinator: Dr. Siddesh G M

Course Content

UNIT-I

Network layer: **Logical addressing** - IPV4 addresses, Address space, notations, classful and classless addressing with problem solving, NAT, IPV6 addresses; IPV6 addresses; **Network layer: Internet protocol** - IPV4 datagram, fragmentation, checksum and options; IPV6 packet format, advantages and extension headers; Transition from IPV4 to IPV6.

UNIT-II

Address mapping, Error reporting, & Multicasting - Address mapping, ARP, RARP, BOOTP and DHCP; ICMP, IGMP, **Network layer: Delivery, Forwarding, & Routing** – Direct Vs Indirect delivery, Forwarding Techniques, Forwarding Process, Routing Table,; **Unicast routing protocols with problem solving** – Optimization, Intra and Inter domain routing, distance vector routing, link state rout routing, path vector routing. **Multicast routing protocols** – Introduction, applications, unicast routing vs multicast routing, source based tree routing, group shared tree routing, multicast distance vector.

UNIT-III

Transport Layer - Process-to-Process delivery, User Datagram Protocol, Transmission Control Protocol, SCTP-services, features. **Congestion control & QOS** - Data traffic, Congestion, Congestion control, Two examples – congestion control in TCP and Frame Relay, Quality of Service, Techniques to improve QOS, **Case study:** QoS requirements in portable devices.

UNIT-IV

Application Layer: Domain Name System - Namespace, Domain name space, Distribution of Name space, DNS in internet, Resolution; **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.

UNIT-V

Cryptography (with problem solving) - Introduction, Symmetric Key cryptography - Traditional ciphers, simple modern ciphers and modern round ciphers, Asymmetric Key cryptography – RSA and Diffie Hellman, **Network Security -** Security Services, Message confidentiality, Message integrity, Message Authentication, Digital Signature, Entity Authentication

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. Wayne Tomasi, Introduction to Data Communications and Networking, Pearson Education, 2005.

Course Outcomes (COs):

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

1. Solve the problems associated with IPV4 and IPV6 addressing. (PO-1,2) (PSO-2)
2. Use appropriate protocols to make routing decisions (PO-1,2,3,5,9,10,12) (PSO-2)
3. Understand different transport layer protocols and various techniques to improve QoS. (PO-1,5,9,10,12) (PSO-2)
4. Describe the working principles of various application layer protocols. (PO-1) (PSO-2)
5. Illustrate principles of network security and Cryptography. (PO-1,2,3,5,9,10,12) (PSO-2)

DATABASE MANAGEMENT SYSTEMS

Course Code: IS52   **Credit: 3:1:0**
Prerequisite: Nil   **Contact Hours: 42L + 14T**
Course Coordinator: Mrs. Lincy Mathews

Course Content

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.

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, Relational Database Design Using ER- to-Relational Mapping.

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

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.
Introduction to MongoDB: Overview, Advantages, Database Modeling, Basic Database Querying Commands

Unit–V
Introduction to Transaction Processing, Transaction and System Concepts, Desirable properties of Transactions, Transaction Support in SQL. Two-phase Locking Techniques for Concurrency Control, Recovery techniques based on Deferred Update, Recovery techniques based on Immediate Update, The ARIES recovery algorithm.

Tutorial Exercises:

1. State the aim and description for any database application.

2. Description of entities, attributes and constraints for the chosen database application.
3. Construct the Entity Relationship diagram for the chosen database application. Document all assumptions that are made about the mapping constraints.
4. Use the ER-to-Relational mapping algorithm to map the ER-diagram into a relational database design for the chosen database application.
5. Continued…
6. Problems on Relational Algebra: Projection, Selection, Set Theory, JOIN and DIVISION, Additional Relational Operations.
7. Normalize the constructed Relational Schema: 1NF, 2NF, 3NF and BCNF
8. Simple Queries on SQL: Select, From, Order By, Group By and Having
9. Complex queries on SQL: Sub Queries, Joins, Exists, Not Exist and Views.
10. Problems using basic constructs of PLSQL: if, if else, while, for.
11. Problems using basic constructs of PLSQL: Procedures, Cursors, Triggers.
12. Simple Queries on MongoDB: Create database, Drop database, create collection, and Drop collection Alter.
13. Simple Queries on MongoDB: Insert Update, Querying Database, and Projection.
14. Simple Queries on MongoDB: Sorting Database, Aggregation.

Text Books:

1. Ramez Elmasri and Shamkant B.Navathe: Fundamentals of Database Systems, 5th Edition, Addison-Wesley, 2007.
2. Benjamin Rosenzweig, Elena Silvestrova Rakhimov: Oracle PL/SQL by Example, 4th Edition, 2010.

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 (COs):

At the end of the course, students will be able to-
1. Design Entity Relationship model for the given database problem. (PO-1,2,3,9,10,12) (PSO-1,3)
2. Develop relational model from the given Entity Relationship diagram. (PO-1,2,3,5,9,10,12) (PSO- 1,3)
3. Design the relational database schema by applying normalization techniques. (PO-1, 2,3,9, 10,12) (PSO- 1, 3)
4. Apply query language to perform database operations. (PO-1, 2,3,5,9, 10,12) (PSO- 1, 3)
5. Understand the concepts of transaction processing, concurrency control and recovery techniques. (PO-1, 12) (PSO- 1,3)

SOFTWARE ENGINEERING

Course Code: IS53   **Credit: 3:0:0**
Prerequisite: Nil   **Contact Hours: 42L**
Course Coordinator: Mrs. Savita K Shetty

Course Content

UNIT-I

Introduction: Professional software development, Software engineering ethics, Case studies. **Software processes:** Software process models, Process activities, coping with change, The Rational Unified process. **Agile Software Development:** Agile methods, Plan-driven and agile development, Extreme programming, Agile project management, Scaling agile methods.

UNIT-II

Requirements engineering: Functional and Non-functional requirements, the software requirements document, Requirements specification, Requirements Engineering Processes, Requirements elicitation and analysis, Requirements validation, Requirements management.

UNIT-III

Architectural Design: Software Design and Implementation, Architectural design decisions, Architectural views, Architectural patterns, Application architectures**. Design and implementation**: Object-oriented design using the UML, Design patterns, Implementation issues, Open source development.

UNIT-IV

Software testing: Development testing, Test-driven development, Release testing, User testing. **Software evolution:** Evolution processes, Program evolution dynamics, Software maintenance, Legacy system management

UNIT-V

Software Management: Project management: Risk management, Managing people, Teamwork. Project planning: Software pricing, Plan-driven development, Project scheduling, Agile planning, **Quality management:** Software quality, Software measurement and metrics. **Process improvement:** The process improvement process, The CMMI process improvement framework.

Text Book:

1. Ian Sommerville, Software Engineering, 9th Edition, Pearson Education, 2011.

References:

1. Roger S. Pressman, Software Engineering-A Practitioners approach, 8th Edition, McGraw-Hill, 2014.
2. Shari Lawrence Pfleeger, Joanne M. Atlee, Software Engineering Theory and Practice, Third Edition, Pearson Education, 2006.
3. Waman S Jawadekar, Software Engineering Principles and Practice, Tata McGraw Hill, 2004.
4. Douglas Bell, Software Engineering for Students, A Programming Approach, 4th Edition, Pearson Education.

Course Outcomes (Cos):

1. Understand the concepts of software engineering and development processes. (PO-1,8,9,10,11,12) (PSO-1)
2. Analyze the functional and non-functional requirements for the given problem. (PO-1,2,9,10,11,12) (PSO-1)
3. Apply software architectural design for the given scenario. (PO-1,2,3,9,10,11,12) (PSO-1
4. Understand Software testing and evolution processes. (PO-1,9,10,11,12) (PSO-1)
5. Analyze Software Project Management issues and process improvement. (PO-1,2,11) (PSO-1)

INTELLECTUAL PROPERTY RIGHTS

Course Code: IS54   **Credit: 3:0:0**
Prerequisite: Nil   **Contact Hours: 42L**
Course Coordinator: Mrs. Evangeline D

Course Content

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. Case Study.

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. Case Study.

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. Case Study.

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.

References:

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

Course Outcomes (COs):

At the end of the course, students will be able to-
1. Understand the principles and importance of Intellectual Property Rights. (PO-1,8,9,10,12) (PSO-1)
2. Analyze the significance for obtaining patents, trademarks and copyrights. (PO-1,2,8,9,10,12) (PSO-1)
3. Understand the process of registration and infringement for patents, trademarks and copyrights. (PO-1,8,9,10,12) (PSO-1)
4. Analyze the applicability of Intellectual property rights in the field of Information Technology. (PO-1,8,9,10,12) (PSO-1)

COMPUTER NETWORKS LABORATORY

Course Code: ISL56   **Credit: 0:0:1**

Prerequisite: Data Communications   **Contact Hours: 14P**

Course Coordinator: Mr. Suresh Kumar K R

Course Content

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 implement a chat application, where the Client establishes a connection with the Server. The Client and Server can send as well as receive messages at the same time. Both the Client and Sserver exchange messages

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 source 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. Write a program to achieve Traffic management at Flow level by implementing Leaky Bucket Algorithm.

8. 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.

9. Write a program to implement Go Back N sliding window protocol

10. Write a program to implement Selective repeat sliding window protocol

11. Write a program to implement the Diffie-Hellman Key Exchange algorithm.

12. Write a program for simple RSA algorithm to encrypt and decrypt the data.

Text Book:

1. Behrouz A. Forouzan, Data Communications and Networking, Fourth Edition, Tata McGraw-Hill, 2006.
2. William Stallings, Cryptography and Network security, Principles and Practices, Third Edition, PHI, 2005

Reference books:

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 (COs):

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

1. Design and implement the functionalities of various layers of the OSI model.
   (PO-1,2,3,9) (PSO-1)
2. Implement and analyze the working of different techniques for data handling to ensure secure and error free transmission. (PO-1,2,3,4,9) (PSO-1)

DATABASE MANAGEMENT SYSTEMS LABORATORY

Course Code: ISL57 **Credit: 0:0:1**

Prerequisite: Nil **Contact Hours: 14P**

Course Coordinator: Mrs. Sunitha R S

Exercise-I

Consider an Employee with a social security number (SSN) working on multiple projects with definite hours for each. Each Employee belongs to a Department. Each project is associated with some domain areas such as Database, Cloud and so on. Each Employee will be assigned to some project. Assume the attributes for Employee and Project relations.

      a)   Mention the constraints neatly.

      b)   Design the ER diagram for the problem statement

      c)   State the schema diagram for the ER diagram.

      d)   Create the tables, insert suitable tuples and perform the following operations in SQL

          1. Obtain the details of employees assigned to "Database" project.

          2. Find the number of employees working in each department with department details.

          3. Update the Project details of Employee bearing SSN = #SSN to ProjectNo = #Project_No and display the same.

      e)   Create the table, insert suitable tuples and perform the following operations using MongoDB

          1. List all the employees of Department named #Dept_name.

          2. Name the employees working on Project Number :#Project_No

      f)   Write a program that gives all employees in Department #number a 15% payincrease. Display a message displaying how many employees were awarded the increase.

Exercise-II

Consider the relations: PART, SUPPLIER and SUPPLY. The Supplier relation holds information about suppliers. The attributes SID, SNAME, SADDR describes the supplier. The Part relation holds the attributes such as PID, PNAME and PCOLOR. The Shipment relation holds information about shipments that include SID and PID attributes identifying the supplier of the shipment and the part shipped, respectively. The Shipment relation should contain information on the number of parts shipped.

      a)   Mention the constraints neatly.

      b)   Design the ER diagram for the problem statement

      c)   State the schema diagram for the ER diagram.

      d)   Create the above tables, insert suitable tuples and perform the following operations in

         Oracle SQL:

          1.   Obtain the details of parts supplied by supplier #SNAME.

          2.   Obtain the Names of suppliers who supply #PNAME.

      e)   Delete the parts which are in #PCOLOR.

f) Create the table, insert suitable tuples and perform the following operations using MongoDB
   1. Update the details of parts for a given part identifier: #PID.
   2. Display all suppliers who supply the part with part identifier: #PID.
g) Write a PL/SQL program to copy the contents of the Shipment table to another table for maintaining records for specific part number.

Exercise-III

Consider the relations BOAT, SAILOR and RESERVES. The relation BOAT identifies the features of a boat such as unique identifier, color and a name. The list of sailors with attributes such as SailorID, name, age etc., are stored in the relation SAILOR. The sailors are allowed to reserve any number of boats on any day of the week and the records are to be updated in the RESERVES table.

a) Mention the constraints neatly.
b) Design the ER diagram for the problem statement
c) State the schema diagram for the ER diagram.
d) Create the tables, insert suitable tuples and perform the following operations in SQL:
   1. Obtain the details of the boats reserved by '#Sailor_Name'.
   2. Retrieve the BID of the boats reserved necessarily by all the sailors.
   3. Find the number of boats reserved by each sailor. Display the Sailor_Name along with the number of boats reserved.
e) Create the table, insert suitable tuples and perform the following operations using MongoDB.
   1. Obtain the number of boats obtained by sailor :#Sailor_Name
   2. Retrieve boats of color :"#color"
f) Write a PL/SQL program to check whether a given number is prime or not.

Exercise-IV

Consider the Banking database – CUSTOMER, BRANCH, ACCOUNT and TRANSACTION. An account can be a savings account or a current account. Customer can have both types of accounts. The transactions can be a deposit or a withdrawal. Mention the constraints neatly.

a) Design the ER diagram for the problem statement
b) State the schema diagram for the ER diagram.
c) Create the above tables, insert suitable tuples and perform the following operations in
   SQL:

   1. Obtain the details of customers who have both Savings and Current Account.
   2. Retrieve the details of branches and the number of accounts in each branch.
   3. Obtain the details of customers who have performed at least 3 transactions.
   4. List the details of branches where the number of accounts is less than the average number of accounts in all branches.

    d) Create the table, insert suitable tuples and perform the following operations using MongoDB
        1. Find the branch name for a given Branch_ID.
        2. List the total number of accounts for each customer.
    e) Using cursors demonstrate the process of copying the contents of one table to a new table.

Exercise-V

Consider the Book Lending system from the library- BOOKS, STUDENT,BORROWS. The students are allowed to borrow any number of books on a given date from the library. The details of the book should include ISBN, Title of the Book, author and publisher. All students need not compulsorily borrow books.

    a) Mention the constraints neatly.
    b) Design the ER diagram for the problem statement
    c) State the schema diagram for the ER diagram.
    d) Create the above tables, insert suitable tuples and perform the following operations in SQL:

        1. Obtain the names of the student who has borrowed either book bearing ISBN '123' or ISBN '124'.
        2. Obtain the Names of female students who have borrowed "Database" books.
        3. Find the number of books borrowed by each student. Display the student details along with the number of books.

    e) Create the table, insert suitable tuples and perform the following operations using MongoDB
        1. Obtain the book details authored by "author_name".
        2. Obtain the Names of students who have borrowed "Database" books.
    f) Write a PL/SQL procedure to print the first 8 Fibonacci numbers and a program to call the same.

Text Book

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

Course Outcomes (COs):

1. Develop relational database schema by using data definition language utilities.
. (PO-1,2,3,4,5,9,10) (PSO-1,3)
2. Develop solutions to the broad range of query and data update problems using SQL/MongoDB, (PO-1,2,3,4,5,9,10) (PSO-1,3)
3. Implement programmatic components for a database using PLSQL. (PO-1,2,3,4,5,9, 10) (PSO-1,3)
4. Analyze the results and produce substantial written documentation (PO-1,2,4,9,10) (PSO-1)

SCRIPTING LANGUAGES LABORATORY

Course Code: ISL58   **Credit: 0:0:2**

Prerequisite: Object Oriented Programming using Java Laboratory Contact Hours: 28P

Course Coordinator: Mrs. Evangeline D.

PART – A

1. **Python:** Write Python code to do the following:
    i. Create list with inputs from user
    ii. Determine minimum and maximum elements in the list
    iii. Insert new element into the list
    iv. Delete an element from the list
    v. Determine if an element is present in the list.

2. **HTML and Javascript:** Create a front end that allows the user to enter details of the item purchased (item name, item price (option), number of items bought). On clicking of the submit button, the total cost for the item purchased should be calculated and displayed. On clicking on the 'paid' button, appropriate alert box should be popped up.

3. **Python:** Write a python program to create a class 'Rectangle'. This class should include a constructor to initialize the dimensions. Include a function in the class to compute the area of the rectangle. Create objects of the class and print area.

4. **Python:** 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).

5. **Javascript:** The function Change String(str) needs to modify the string passed using the following rules:
    * Replace every letter in the string with the letter that follows it in alphabetical order (ie. Z becomes a, l becomes m).
    * Take every vowel in this new string (a, e, i, o, u) and Capitalize it.
    * Return this modified string.

6. **Python:** Write Python code to do the following operations:
    * Create a dictionary that contains the atomic element symbol and its name.
    * Add a unique & duplicate element into this dictionary by interacting with the user. Observe the output and justify it.
    * Display the number of atomic elements in this dictionary
    * Ask user to enter an element to search in the dictionary. Display appropriate results.

    Rewrite this program so that these operations are inside a function called 'AtomicDictionary'. Create another python file called "Atomic.py" and execute this function in it.

7. **Python:** Create a Python class called 'Student' having 'name', 'age' as attribute along with a list having the marks obtained for three subjects.
    * Create a constructor to initialize two objects of this class.
    * Create a member function called 'display' printing the details of a specific object
    * Ask user to enter the values for an object through an 'accept' member function.
    * Display these details.

8. **Python:** Create a list of 6 numbers. Use 'list-comprehension' to create a new list where each element in the original list is multiplied by 3. Use 'lambda' and 'reduce' function find the sum of the elements of the original list as well as the new list.

9. **Javascript:** The function HourMinute(num) must return the number of hours and minutes the parameter converts to. Separate the number of hours and minutes with a colon. The function AgeConvert(num) needs to calculate the age of a person, given the birth date.

10. **Javascript:** The function LetterSurround(str) needs to find out if the string passed is an acceptable sequence by either returning true or false. The string parameter will be composed of + and = symbols with several characters between them. For the string to be true each letter must be surrounded by + symbol. Test Cases: The string will not be empty and will have at least one letter.

11. **JavaScript – Client Side Validation:** Design a case study for a Bakery that creates and validates a HTML form at the client side using Javascript. Bakery Menu & Price calculation of items bought should be the result. Perform the necessary Client Side validation using JavaScript.

12. **Javascript and JSON:** Create the details (name, native country, story title and publication year) of four authors as a JSON object. Display a web page displaying the details of
    - First two authors in a tabular form using HTML Table Tag.
    - Other two authors as plain text.

PART – B

1. **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.

2. **Python File Handling & List Comprehension:** Write a python program to read the contents of a file (filename as argument) and store the 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.

3. **Python for Data Science:** Load the Titanic dataset into one of the data structures (NumPy or Pandas) and perform the following operations.
    a. Display header rows and description of the loaded dataset.
    b. Remove unnecessary features (E.g. drop unwanted columns) from the dataset.
    c. Manipulate data by replacing empty column values with a default value.
    d. Perform the following visualizations on the loaded dataset:
        - Passenger status (Survived/Died) against Passenger Class
        - Survival rate of male vs female
        - No of passengers in each age group

4. **Python for Data Science:** Load the 'Student Performance' dataset into one of the data structures (NumPy or Pandas) and perform the following operations.
    a. Display header rows and description of the loaded dataset.
    b. Remove unnecessary features (E.g. drop unwanted columns) from the dataset such as 'lunch' and 'test preparation course .

c. Convert the attribute 'race/ethnicity' to have 'groupA' to be 'Asian Students', 'groupB' to be 'African Students' , 'groupC' to be 'Afro-Asian Students', 'groupD' to be 'American Students' and 'groupE' to be 'European Students'.

d. Perform the following visualizations on the loaded dataset:
   - Tally of the Number of Male & Female students who took up the 'test preparation course' and those who did not.
   - Total Number of Male & Female Students belonging to each student group
   - No of students who 'failed'(less than 40), 'second class'(between 40 & 50), 'first class'(between 60 & 75) and 'distinction'(above 75) in 'Maths', 'Reading' and 'Writing'.

5. **Python for Data Science -** Perform Data Visualization on Iris Dataset

a)Load the Titanic dataset into one of the data structures (NumPy or Pandas).

b)Display header rows and description of the loaded dataset.

c) Clean the data if applicable

d) Find the average petal width of each category of IRIS Species

e) Data Visualization for:

(i) How many flowers of each species exists for each value of sepal width

(ii) How many flowers are there whose petal width is <1, between 1 to 2 and >2

(iii) Tally the Iris-Versicolour and Iris-Virginica species according to the value of Sepal Width

6. **Javascript:** Design a HTML page with two textboxes and button. Let user enter a sentence in the first text box and the search string in the second. Use Javascript to perform the following operations on clicking the button
   a. Display the entered text
   b. Display the number of digits, white space characters and alphabets in the entered text
   c. Check the number of occurrences of any given word.
   d. Determine the position of occurrence of the search string in the given string

7. **Javascript:** Design a HTML page to accept a string in a text box on the click of a button. Use Javascript to perform the following operations
   a. Add the string received to a pre-existing Javascript array named "Original_String" at its beginning
   b. For each array element, use a callback function and store each one's length in a new array named "MyLength"
   c. Write a Javascript function that takes the arrays "Original_String" and "MyLength" as argument. This function filters those strings with length less than or equal to 3.

8. **JavaScript (JSON):** Create two JSON objects. One contains the details of a 'Patient' as "name", "AadharNumber" and a JSON array which has the "lab-tests" the patient has taken. The other contains the 'Hospital' details as "hospital-name" and "location". Create a web page that displays the Hospital details when the page loads along with the text "Patient Details:" On mouse-hover the text "Patient Details:" changes colour and displays the details of the patient stored in the JSON object.

9. **Python and JavaScript – File Reading:** Write a Python code that demonstrates the following:
   a. Sending of a File name from back a response from Python Backend to HTML
   b. Extract file contents to HTML from backend python code

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
   a. Negative amount cannot be entered and Users cannot withdraw more than 5000 at one time
   b. 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 and JavaScript - Student Registration Application:** Design a HTML form that displays: Two text fields t to input the user's USN and Date of Birth and Three text boxes to input three marks. Perform the following operations.
   a. Validate the data entry on the client side using Javascript so that null values are not accepted for all the five text boxes.
   b. Validate the entry on server-side using Python to ensure that USN is accepted in a proper pattern as well as date validations are done. Calculate the average using Python on server-side and display the result.

Text Book

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

Course Outcomes (COs):

At the end of the course, student will be able to -
1. Use the concepts of Python and Java scripts to develop solutions for the given problems. (PO-1,2, 3,4,9,10) (PSO-3)

2. Develop dynamic web applications to respond user driven events and client side validations. (PO-1,2,3,4,9,10) (PSO-1)

3. Analyze the results and produce substantial written documentation (PO-1,2, 4,5,9,10) (PSO-1,3)

DIGITAL IMAGE PROCESSING

Course Code: ISE551   **Credit: 3:0:0**
Prerequisite: Nil   **Contact Hours: 42L**
Course Coordinator: Dr. Megha P Arakeri

Course Content

UNIT-I
Introduction: Digital image processing, Imaging modalities, Fundamental steps in image processing, 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 (Histogram equalization, Histogram matching), Fundamentals of spatial filtering (Mechanics of spatial filtering, correlation and convolution), Smoothing spatial filters, Sharpening spatial 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), Region descriptors (simple descriptors, topological descriptors, texture).

UNIT-IV
Object Recognition: Patterns and pattern classes, Recognition based on decision-theoretic methods, Matching, Optimum statistical classifier, Neural networks. **Image Compression:** Fundamentals, Compression methods (Huffman, Arithmetic, Run-length coding)

UNIT-V
Color Image Processing: Color models, Full-color image processing. **Morphological Image Processing:** Erosion and Dilation, Opening and closing, Hit-or-miss transform, Morphological algorithms (Boundary extraction, Hole filling, Extraction of connected components).

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 (COs):

1. Describe the fundamental concepts of a digital image processing (PO-1) (PSO-3).
2. Apply spatial domain filters to improve the quality of the image (PO-1,2,3,5) (PSO-3).
3. Use segmentation, description, compression, and recognition techniques in processing the image. (PO-1,2,3,5) (PSO-3).
4. Apply colour models and morphological operations in image processing. (PO-1,2,3,5) (PSO-3).
5. Design solution for the identified problem by applying appropriate image processing algorithm (PO-1,2,3,5,9,10,11,12) (PSO-3).

NATURAL LANGUAGE PROCESSING

Course Code: ISE552   **Credit: 3:0:0**
Prerequisite: Nil   **Contact Hours: 42L**
Course Coordinator: Ms. Rajeshwari S B

Course Content:

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 (COs):

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

1. Illustrate how speech and language technology relies on formal models to capture Knowledge. (PO-1) (PSO-3)
2. Apply N-gram and part-of-speech tagging techniques for word prediction, spell check and pronunciation processing. (PO-1,2, 3,5,9,10,11,12) (PSO-3)
3. Use unification and early algorithms for capturing syntactic information. (PO-1,2,3,5,9,10,11,12) (PSO-3)
4. Apply probabilistic context free grammar and first order predicate calculus to represent meaning. (PO-1,2,3,5,9,10,11,12) (PSO-3)
5. Understand the issues of semantic analysis, lexical semantics & discourse. (PO-1) (PSO-3)

INTERNET OF THINGS

Course Code: ISE553	**Credit: 3:0:0**
Prerequisites: Internet of Things	**Contact Hours: 42**
Course Coordinators: Dr. Yogesh H K/ Mr. Jagadeesh Sai D	

Course Content

UNIT-I

Introduction to Internet of Things Definition & Characteristics of IoT, Physical Design of IoT Things in IoT, IoT Protocols, Logical Design of IoT, IoT Functional Blocks, IoT Communication Models, IoT Communication APIs, IoT Enabling Technologies , Wireless Sensor Networks, Cloud Computing Big Data Analytics ,Communication Protocols , Embedded Systems IoT Levels & Deployment Templates, IoT Level-1,IoT Level-2,IoT Level-3,IoT Level-4,IoT Level-5,IoT Level-6

UNIT-II

IoT and M2M: Introduction, M2M, Difference between IoT and M2M, SDN and NFV for IoT, Software Defined Networking, Network Function Virtualization, IoT System Management with NETCONF-YANG, Need for IoT Systems Management, Simple Network Management Protocol (SNMP), Limitations of SNMP, Network Operator Requirements, NETCONF, YANG IoT Systems Management with NETCONF-YANG, NETOPEER.

UNIT-III

IoT Platforms Design Methodology: IoT Design Methodology , Purpose & Requirements Specification , Process Specification, Domain Model Specification, Information Model Specification , Service Specifications, IoT Level Specification, Functional View Specification, Operational View Specification, Device & Component Integration, Application Development, **IoT Systems** - Logical Design using Python ,Functions Modules ,Packages ,File Handling Operations Classes, Python Packages of Interest for IoT ,JSON, XML, HTTPLib & URLLib ,SMTPLib

UNIT-IV

Raspberry Pi, About the Board, Linux on Raspberry Pi, Raspberry Pi Interfaces, Serial SPI, I2C, Programming Raspberry Pi with Python, Controlling LED with Raspberry Pi , Interfacing an LED and Switch with Raspberry, Interfacing a Light Sensor (LDR) with Raspberry Pi ,Other IoT Devices, pcDuino, Beagle Bone Black, Cubie board. IoT Physical Servers & Cloud Offerings, WAMP - AutoBahn for IoT, Xively Cloud for IoT, Python Web Application Framework – Django, Django Architecture, Starting Development with Django, Designing a RESTful Web API, Amazon Web Services for IoT, Amazon EC2, Amazon AutoScaling, Amazon S3, Amazon RDS Amazon DynamoDB, Amazon Kinesis, Amazon SQS, Amazon EMR, SkyNet IoT Messaging Platform, INTEL Gen2, UDDO Board example.

UNIT-V

Data Analytics for IoT, Apache Hadoop, Map Reduce Programming Model, Hadoop Map Reduce Job Execution, Map Reduce Job Execution Workflow, Hadoop Cluster Setup, Using Hadoop Map Reduce for Batch Data Analysis, Hadoop YARN, Apache Oozie, Setting up Oozie, Oozie Workflows for IoT Data Analysis, Apache Spark, Apache Storm, Setting up a Storm Cluster, Using Apache Storm for Real-time Data Analysis, REST-based approach, Web Socket-based approach.

Text Book:

1. Internet of Things (A Hands-on-Approach) by Arshdeep Bagha, Vijay Madisetti University press 2015.

Reference:

1. **Enterprise IoT:** Strategies and Best Practices for Connected Products and Services By Dirk Slama, Frank Puhlmann, Jim Morrish, Rishi M Bhatnagar

Course Outcomes (COs):

At the end of the course, student will be able to -
1. Understand the design issues and fundamentals of IoT (PO-1,9,10,12) (PSO-2,3)
2. Design various methodologies for M2M and SDN architectures. (PO-1,9,10,12) (PSO-2,3)
3. Distinguish different cloud based solution for IoT (PO-1,2,9,10,12) (PSO-2,3)
4. Develop the IoT based solutions for the real world problems. (PO-1,2,3,4,5,6,9,10,12) (PSO-2,3)
5. Analyze the various data analytical tools in IoT (PO-1,2,9,10,12) (PSO- 2,3)

DEVELOP IN SWIFT FUNDAMENTALS

Course Code: ISE554   **Credit: 3:0:0**
Prerequisite: Nil   **Contact Hours: 42L**
Course Coordinator: Mr. Shashidhara H S

Course Content

UNIT-I
Getting Started with App Development - basics of data, operators, and control flow in Swift, debugging, Xcode, building, running and debugging an app, Interface Builder. Guided project - Light.

UNIT-II
Introduction to UI Kit - Swift strings, functions, structures, collections, and loops. UIKit—the system views and controls that make up a user interface and display data using Auto Layout and stack views. Guided project - Apple Pie

UNIT-III
More Swift – Collections, Structures, Classes, Closures

UNIT-IV
Navigation and Workflows - build simple workflows and navigation hierarchies using navigation controllers, tab bar controllers, and segues, optionals and enumerations. Guided project - Personality Quiz

UNIT-V
Tables and Persistence - scroll views, table views, and building complex input screens, save data, share data to other apps, work with images in the user's photo library. Guided project - List, a task-tracking app that allows the user to add, edit, and delete items in a familiar table- based interface.

Text Book:

1. Develop in Swift Fundamentals, Apple Books

Course Outcomes (COs):

At the end of the course, students will be able to
1. Apply swift programming constructs to solve the given problem (PO-1,2,3,5) (PSO-1)
2. Design user interfaces for an iOS app using UIKit framework (PO-1,2,3,5) (PSO-1)
3. Apply advanced constructs of swift to solve the given problem. (PO-1,2,3,4,5) (PSO-1)
4. Develop iOS apps using segues and navigation (PO-1,2,3,4,5) (PSO-1)
5. Design views for an iOS app using table view and scroll views controllers (PO-1,2,3,4,5) (PSO-1)

SOFTWARE PROJECT MANAGEMENT

Course Code: ISOE06 **Credit: 3:0:0**
Prerequisite: Nil **Contact Hours: 42L**
Course Coordinator: Dr Sumana M

Course Content

UNIT-I
Introduction, Contract & Technical project Management, Activities, Plans, Methods, Methodologies, objectives, business case, Success, failure, Management control, Traditional vs Modern project management, Project portfolio management

UNIT-II
Project evaluation, Cost-benefit evaluation Techniques, Risk Evaluation, Resource allocation, Strategic management, Benefits, Step Wise Project Planning. Build/Buy? Methodologies, software processes, process models, prototyping, Incremental delivery, Atern, RAD, Agile methods, XP, Scrum, Selection of process model.

UNIT-III
Basis for software effort estimation, models, Expert judgment, Estimation by analogy, Albrecht FPA, FP Mark II, COSMIC FFP, COCOMO II, Cost estimation, Staffing pattern, Schedule compression, Capers Jones rules, When activity planning? Project schedules & activities, Sequencing & scheduling activities, Network Model, Time, Forward & Backward Pass, Critical path, Activity float, Shorten project duration, Critical activities, Activity on Arrow networks.

UNIT-IV
Categories of risk, deal with risk, Risk identification, assessment, planning, Management. Evaluation of risks to the schedule, PERT, Critical chain. Nature of Resources, resource requirements, scheduling resources, creating critical paths, Counting the cost, Publish resource schedule, cost schedules, scheduling sequence.

UNIT-V
Monitoring and control Framework, collect data, Review, Project termination, progress, cost monitoring, Earned Value Analysis, Prioritizing Monitoring, get project back to target, change control, Software Configuration Management, Managing contracts, Stages, terms of contract, contract management, Acceptance.

Text Book:

1. Bob Hughes, Mike Cotterell, Software Project Management, 4th Edition, Tata McGraw Hill Publications, 2006.
2. Kathy Schwalbe, Information Technology Project Management, 5th Ed, Thompson, 2006.
3. Watts S. Humphrey, Managing the Software Process, Addison-Wesley, 1989.

Course Outcomes (COs):

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

1. Recognize the importance of software project management. (PO-1,2)
2. Classify and evaluate projects with illustrations. (PO-1,2,3,4)
3. Select a project approach, estimate effort and plan activities by analysis and evaluation. (PO-1,3,4)
4. Analyze, evaluate and manage risks and allocate resources. (PO-1,3,4)
5. Analyze, evaluate, monitor and control projects and manage contracts. (PO-1,3,4)

OBJECT ORIENTED PROGRAMMING WITH C++

Course Code: ISOE07   **Credit: 3:0:0**
Prerequisite: CS26, Fundamentals of Computing   **Contact Hours: 42L**
Course Coordinator: Dr. Yogish H K

Course Content

UNIT-I

Introduction to C++: Procedure-Oriented Programming Systems, Object-Oriented Programming Systems, Comparison of C++ with C, Sample C++ program. Console Input/Output in C++, Variables in C++, Reference Variables in C++, Function Prototyping, argument passing, Function Overloading, Default Values for Formal Arguments of Functions, Inline Functions.

UNIT-II

Class and Objects: Review of structure - Defining a structure, declaring structure variables, accessing structure members. Sample programs using structure. Introduction to Classes and Objects, Difference between structure and classes. Member Functions and Member Data, Member function overloading, Inline functions, Static class members, returning objects, Array of objects, passing objects, Friend functions.

UNIT-III

Dynamic Memory Management: Introduction, Dynamic Memory Allocation, Dynamic Memory Deallocation.
Constructors and destructors: Introduction, default constructors, parameterized constructors, copy constructors, Destructors, overloaded constructors. Sample programs.
Templates: Introduction, Function Templates, more than one template type parameters, over loading and Overriding a Function template, template Class, more than one template (Generic) type parameters.

UNIT-IV

Inheritance: Introduction, Base class access control, inheritance and protected members, types of Inheritances, constructors, destructors, passing arguments to base class constructor, granting access, Virtual base classes. Virtual functions.

UNIT-V

Operator Overloading: Operator Overloading, Overloading the Various Operators – Overloading the Increment and the Decrement Operators (Prefix and Postfix), Overloading the Arithmetic Operators and relational operators. Overloading the Insertion and Extraction Operators. **Exception Handling:** Exception handling fundamentals, catching all exceptions.

Text Book:

1. Object Oriented Programming With C++ 7Th Edition, Tata McGraw-Hill, 2017
2. Herbert Schildt, The Complete Reference C++, 4th Edition, Tata McGraw-Hill, 2005.

Reference Book:

1. Sourav Sahay, Object Oriented Programming Using C++ ,Sourav Sahay, 2nd edition, 2013

Course Outcomes (COs):

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

1. Understand the need of using Object Oriented Programming in the real-world applications using reference variables, and various functions. (PO-1,2) (PSO1)
2. Develop programs using classes and objects. (PO-1,2) (PSO1)
3. Demonstrate dynamic memory allocation using new and delete, design a program using Templates and Exception Handling. (PO -1,2,3) (PSO1).
4. Develop the mechanism of deriving a new class from older classes through inheritance. (PO-2,3)(PSO-1)
5. Constructing applications to provide flexible options for the creation of new definitions for some of the operators. (PO-2,3) (PSO1)

MACHINE LEARNING

Course Code: IS61 Credit: 3:1:0
Prerequisites: Scripting Languages Contact Hours: 42L + 14T
Course Coordinator: Dr. Mydhili K Nair

Course Content

UNIT-I
Machine Learning Introduction: Learning, Types of Machine Learning, Types of Machine Learning, Supervised Learning, The Machine Learning Process.
Machine Learning Preliminaries: Terminology - Weight Space, The Curse of Dimensionality; Testing Machine Learning Algorithms – Over-fitting, Training, Testing and Validation Sets, The Confusion Matrix, Accuracy Metrics, ROC Curve, Unbalanced Dataset, Measuring Precision.
Turning Data into Probabilities: Minimizing Risk, maximum a posteriori hypothesis; Basic Statistics: Averages, Variance and Covariance, The Gaussian; Bias-Variance Trade-off

UNIT-II
Neurons, Neural Networks and Linear Discriminants: The Brain and the neuron, neural networks, The Perceptron, Linear Separability.
Multi-layer Perceptron - 1: Going Forwards, Going Backwards – Back Propagation of Error - The Multi-Layer Perceptron Algorithm, Initialising the Weights - Examples as Numerical, Different Output Activation Functions, Sequential and Batch Training.
Multi-layer Perceptron -2: Local Minima, Picking-Up Momentum, Mini-batches and Stochastic Gradient Descent, Other Improvements.
Multi-layer perceptron in practice: Amount of Training Data, Number of Hidden Layers, when to Stop Learning.

UNIT-III
Supervised Learning: Learning a Class from Examples, Linear, Non-linear, Multi-class and Multi-label classification.
Supervised Learning Algorithms - Decision Trees: ID3 - Technique, Examples as Numerical; Classification and Regression Trees (CART)
Supervised Learning Algorithms - Regression: Linear Regression - Technique, Examples as Numerical; Multiple Linear Regression, Logistic Regression Examples as Numerical.
Dimensionality Reduction - Supervised: LDA (Linear Discriminant Analysis) -Technique, Examples as Numerical.

UNIT-IV
Supervised Learning - K-Nearest Neighbour Methods - Technique, Examples as Numerical
Probabilistic Learning: The Naive Bayes Classifier -examples and numerical
Support Vector Machines: Linear and Nonlinear - Technique, Examples as Numerical; Kernel Functions

UNIT-V

Unsupervised Learning: Partitional Clustering - K-Means, Dealing with noise, Examples as Numerical, Elbow method to choose the right value of 'k', problems with k-means clustering;

Unsupervised Learning: Hierarchical Clustering – Agglomerative (AGNES), Divisive(DIANA), examples as numericals

Dimensionality Reduction - Unsupervised: Introduction, Subset Selection, PCA (Principal Component Analysis) – Technique, Examples as Numerical.

Tutorial - Numerical Examples On:

1. Confusion Matrix and Accuracy Metrics

2. Multi-layer Perceptron

3. Decision Tree

   a. ID3 - Classification

   b. CART - Classification

   c. CART - Regression

4. Linear Regression

5. Logistic Regression

6. K Nearest Neighbour Classifier

7. Naive Bayesian Classifier

8. Support Vector Machine

9. Partitional Clustering: K-Means

10. Hierarchical Clustering - Agglomerative and Divisive

11. Dimensionality Reduction Supervised - LDA

12. Dimensionality Reduction UnSupervised - PCA

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", CBSPublishers & Distributors, 2010.
2. Mehryar Mohri, Afshin R, Ameet Talwalkar, "Foundations of MachineLearning", MIT Press, 2012.
3. Charu C. Aggarwal, "Data Classification Algorithms and Applications",CRC Press, 2014
4. Charu C. Aggarwal, "Data Clustering Algorithms and Applications", CRCPress, 2014.
5. Kevin P. Murphy "Machine Learning: A Probabilistic Perspective", TheMIT Press, 2012

Course Outcomes (COs):

At the end of the course, student will be able to -

1. Analyze the various performance metrics used in machine learning. (PO-1,2,6,9,10,12) (PSO-3)
2. Apply neural networks for various scenarios.. (PO-1,2,3,6,9,10,12) (PSO-3)
3. Use supervised learning methods for Classification and Regression (PO-1,2,3,6,9,10,12 (PSO-3)
4. Apply supervised learning techniques and dimensionality reduction using LDA. (PO-1,2,3,6,9,10,12) (PSO-3)
5. Apply unsupervised learning techniques and dimensionality reduction using PCA (PO-1,2,3,6,9,10,12) (PSO-3)

OBJECT ORIENTED ANALYSIS AND DESIGN PATTERNS

Course Code: IS62   **Credit: 3:0:0**
Prerequisite: Object Oriented Programming using Java.   **Contact Hours: 42L**
Course Coordinator: Mrs. Ashwitha

Course Content:

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

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 (COs):

At the end of the course, students will be able to-
1. Design solution to the given problem definition using UML notations (PO-1,2,3,5,7,9,10,11,12) (PSO-1)
2. Apply Façade and Adapter patterns to given problem with traditional design (PO-1,2,3,5,7,9,10,11,12) (PSO-1)
3. Apply Strategy, Bridge and Abstract Factory patterns to the given scenario. (PO-1,2,3,5,7,9,10,11,12) (PSO-1)
4. Apply Template Method, Observer and Decorator patterns for a given problem. (PO-1,2,3,5,7,9,10,11,12) (PSO-1)
5. Use Singleton,Double checked locking pattern and Factory method pattern for a given problem (PO-1,2,3,5,7,9,10,11,12) (PSO-1)

MINI-PROJECT

Course Code: IS65	Credit: 0:0:4
Course Coordinator: Internal Guide	Contact Hours: -

Course Content

Guidelines:

Students have to work in a group of 3/4 to solve a problem in the specific domain. An Internal Guide is allotted per batch based on their domain of expertise 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.

Following are the Rubrics considered for Evaluation of mini-project:

Relevance of Project: Student are expected to clearly state the relevance of project to current IT environment and Society in general

Literature Survey: Student need to study research articles/ existing projects to identify the gaps in the identified problem statement.

Design: Student should prepare design document by considering class/use case/component diagram/state model/sequence model/activity/Interaction model.

Implementation: Student need to implement the designed model using the suitable techniques.

Presentation: Periodically student need to present their progress in front of the evaluation committee. Depending upon the quality of ppt, depth of coverage, answering capabilities to questions raised and division of labor identified during presentation and team work, evaluation committee will be deciding score in this criteria for individual students.

Report: Each group need to prepare the project report and submit the same to the department. Reports need to be adhered to the standard format defined by the department.

Course Outcomes (COs):

 At the end of the course Students will be able to:

1. Identify a problem, review research literature and analyze requirements   (POs-1,12) (PSO-1,2,3)

2. Schedule milestone and deliverables using appropriate project management techniques (POs-8,9,10,11).

3. Design and implement the solution to selected problem using standard models and processes (POs-1,2,3,4,5,6,7,8,9,10,11,12)(PSO – 1,2, 3)

4. Analyze the results and produce substantial written documentation(POs-1,2,4,,8,9, 10,11,12)(PSO-1,2,3)

MACHINE LEARNING LABORATORY

Course Code: ISL66   **Credit: 0:0:1**

Prerequisites: Scripting Languages   **Contact Hours: 14P**

Course Coordinator: Mrs. Pushpalatha M N

Laboratory Experiments:

Implement the following programs using Python

1. **Model Measurement Analysis**: Create a dataset of your choice with at least 10 records. **E.g**. Corona Virus patients who were tested, Student assignments subjected to plagiarism check. Assume sample size of 100. Record the values of TP, TN, FP, FN with varying thresholds set. At each step of varying thresholds calculate the values of Precision, Recall, F1 Score as well as the TPR and FPR. Plot the ROC Curve. Analyze, Interpret.

2. **Artificial Neural Networks - Single Layer Perceptron:** Implement a Single Layer Perceptron using minimal inbuilt functions. Create a dataset containing at least 100 records. Each record should have at least 4 floating point features and a binary label (0 - negative or 1 - positive). Split the dataset into test and train data, initialize the weights, learning rate, epochs and define the activation function. Train the model (Learn the weights of the perceptron on the training data). Print the learned weights and the hyperparameters (epoch and learning rate). Predict the outputs on train and test data. Print the confusion matrix, accuracy, precision, recall on train and test data

3. **Artificial Neural Networks - Multi Layer Perceptron:** Build an Artificial Neural Network by implementing the Back Propagation Algorithm. Test the same using appropriate data sets. Compare the actual and predicted output. Analyze and write the inference.

4. **Supervised Learning Algorithms - Decision Trees:** Implement decision trees considering a data set of your choice.
      (a) Create a ID3 Decision Tree
      (b) Create a CART Decision Tree
      (c) Compare and Contrast the two

5. **Supervised Learning Algorithms - Linear Regression:** Consider a dataset from UCI repository. Create a Simple Linear Regression model using the training data set. Predict the scores on the test data and output RMSE and R Squared Score. Include appropriate code snippets to visualize the model. Interpret the result.

6. **Supervised Learning Algorithms - Logistic Regression:** Implement logistic regression and test it using any dataset of your choice from UCI repository. The output should include Confusion Matrix, Accuracy, Error rate, Precision, Recall and F-Measure.

7. **Supervised Learning Algorithms - KNN:** Implement k-Nearest Neighbor (KNN) by writing the algorithm on your own , without using pre-built code or library, for classifying a dataset. Perform necessary pre-processing steps. Analyse the importance of pre-processing.

8. **Probabilistic Supervised Learning - Naive Bayes:** Create a dataset from the sample given to you(e.g. "Play Tennis Probability", "Shopper Buying Probability" etc.). Perform the necessary pre-processing steps such as encoding. Train the model using Naive Bayes Classifier. Give new test data and predict the classification output. Handcode the

classification probability and compare with the model output. Analyze and write the inference.

9. **Supervised Learning Algorithms - Support Vector Machines:** Generate a separable dataset of size 1000 and 2 features. Plot the samples on a graph and mark the support vectors for the dataset. Also, show that changing the vectors other than the support vectors has no effect on the decision boundary.

10. **Supervised Learning Algorithms - Support Vector Machines:** Use SVM to classify the flowers in Iris dataset. Visualize the results for each of the following combinations:
    (a) For every pair of (different) features in the dataset (there are 4). Which pair separates the data easily?
    (b) Using One-vs-Rest and using One-vs-One. Which one fits better? Which one is easier to compute? Why?
    (c) Using different kernels (Linear, RBF, Quadratic).

11. **Un-Supervised Learning Algorithms - Clustering:** Using any dataset from the UCI repository implement any one type of Hierarchical and Partitional Clustering you are familiar with. Plot the Dendrogram for Hierarchical Clustering and analyze your result. Plot the clustering output for the same dataset using these two partitioning techniques. Compare the results. Write the inference.

12. **Un-Supervised Learning Algorithms - K-Means Clustering:** Build a K-Means Model for the given dataset. In K-Means choosing the K value that gives a better model is always a challenge. We increase the value of K with a dataset having N points, the likelihood of the model increases, and obviously K<N, so to rank or maximize the likelihood we use BIC (Bayesian Information Criterion. Now,
    (a) Build a K-Means Model for the given Dataset (You can use the library functions)
    (b) Implement the BIC function that takes the cluster and data points and returns BIC value
    (c) Implement a function to pick the best K value, that is maximize the BIC.
    (d) Visualize the pattern found by plotting K v/s BIC.

References:

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

Course Outcomes (COs):

At the end of the course, student will be able to -
1. Design the experiment for the given problem using various Machine Learning Algorithms . (PO-1,2,4,5,8,9,11) (PSO-2)
2. Develop the solution for the given real world problem. (PO-1,2,3,4,5,7,8,9,11) (PSO-2)
3. Analyze the results and produce substantial written documentation. (PO-1,3,8,9) (PSO-2)

OBJECT ORIENTED ANALYSIS AND DESIGN PATTERNS LABORATORY

Course Code: ISL67   **Credit: 0:0:1**

Prerequisite: Object Oriented Programming using Java/C++ Laboratory **Contact Hours: 14P**

Course Coordinator: Mrs. Evangeline D

Course Content

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:

   Indoor Sports -   'GamesOnTable' (e.g. Table Tennis, Billiards, Snooker etc.)
                     'BoardGames' (e.g. Carom, Chess etc.)
                     'CourtGames' (e.g. Basketball, Badminton, Kabaddi etc.)

   Outdoor Sports - '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 'get Current Stock' 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
   > do Payment – 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 (COs):

At the end of the course, students will be able to-
1. Design the experiment for the given problem using various Object Oriented Analysis and Design Patterns (PO-1,2,3,9,10) (PSO-1)
2. Develop the solution for the given real world problem (PO-1,2, 3,4,9,10,) (PSO-1)
3. Analyze the results and produce substantial written documentation. (PO-1,2,4,9,10) (PSO-1)

JAVA & J2EE LABORATORY

Course Code: ISL68   **Credit: 0:1:1**

Prerequisite: Object Oriented Programming using Java Laboratory   **Contact Hours: 14P**

Course Coordinator: Mr. Koushik S

Course Content

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 caller 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
      i. To delete the call
      ii. To go on to the next missed call, or
      iii. To display the call details (number, caller name and time). Delete the number if user specifies a number to delete.

   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.
   a. Books collection should include title, author, publisher and price.
   b. Write a method to sort the books in ascending order of price and store it in another List. Maintain the book details with respect to an unique book id.
   c. Prompt for an author name and list all the books with the same author name. Create a new list holding all the book details with price greater than a user specified price.
   d. For a given a value by the user, find all the books that match either the whole or a part of the book title.
   e. Identify a publisher and print books from a particular publisher. Update the publisher details based on a title.

3) Create a desktop java application using swings to enable an user to enter student information such as name, usn, age, address, sgpa of 8 semesters, category.
   a. Perform validations on all the fields. Display appropriate messages in pop up boxes to indicate wrong entries.
   b. On clicking of the "compute" button, find the cgpa . On clicking of the "done" button, mouse place the student details in a collection.
   c. Display the collection in a textarea on the click of a button.

4) Write a java program using Servlet to validate user login information using dialog boxes.
   a. Once validated, allow the user to enter the customer id, if the person is a new customer, else check whether the customer exists in a collection and obtain the customer id.
   b. The customer id can be obtained given a mobile number. Allow the user to enter the item purchased by giving the item id and quantity purchased.

     c. On clicking of a button, the item name and the total cost should appear in the corresponding GUI components.

     d. Using option dialog box, indicate the types of discount available for the customer. On clicking on the print button, print the details in information dialog box.

5) 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. Use appropriate Swing components to insert values in a form.

     b. Use another form to display Representative's information when Credit_Limit is above 15,000.

6) 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, A Radio-button that allows the user to choose between T-Shirt with chest pocket and without. A Combo Box to choose your T-Shirt color, A Button called "Click Me"

     c. Insert the details entered into a table called 'TShirts'.

     d. An Order No is generated by adding '1' to the existing 'OrderNo'

     e. If 'TShirts' table is empty the initial value of 'OrderNo' is 100.

     f. This 'OrderNo' is also inserted into the 'TShirts' table

     g. 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.

7) Create the following application with Struts2 and Hibernate framework

     a. Create a Telephone Directory Application that searches the database based on phone number or name. Also show database table creation with inserting 2-3 values to the table.

     b. Database Name: OnlineDirectory, Table Design: Table Name: Telephone_Directory, Attributes: Phone_Number, Name, Address, Company, Pin_Code.

8) Create the following application with Struts2 and Hibernate framework

     a. Create two tables Flight (Flight_Number, Airline_Name, Weekdays) and seat Reservation (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 Seat Reservation. On Click 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.

Java and J2EE Tutorial

1. java.util Part 1: The Collections Framework: Collections Overview, The Collection Interfaces

2. The Collection Classes: The Array List Class, Linked List Class, Hash Set Class.

3. Accessing a Collection via Iterator, Storing User-Defined Classes in Collections;

4. Working with Maps, Arrays

5. Introduction to Swing, Swing Components,

6. Swing basic containers, Swing Components;
7. Introduction to Servlet: Architecture of Servlet.
8. Basics of Servlet application programming Interface: Servlet Application Programming Interface.
9. The Servlet Architecture, the Servlet Life Cycle.
10. Working with Databases: Types of JDBC Drivers, How to access database, Using Databases, Connecting to Database.
11. Java Threads
12. Struts2 Framework
13. Hibernate Framework
14. Connecting Struts2 and Hibernate (CRUD 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 Subrahmanya, J2EE Architecture, TATA McGRAW-HILL Edition 2007.

Course Outcomes (COs):

At the end of the course, students will be able to-
1. Develop solutions for the given problem using Java and J2EE. (PO-1,2,3,5,6) (PSO-1)
2. Apply java and J2EE concepts to provide solutions in various domains. (PO-1,2,3,4,5,6,9,10,11,12) (PSO-1)
3. Interpret the results and produce the substantial document. (PO-1,2,4,10) (PSO-1)

DEVELOP IN SWIFT DATA COLLECTIONS

Course Code: ISE631   **Credit: 3:0:0**
Prerequisite: iOS App Development   **Contact Hours: 42L**
Course Coordinator: Mr. Shashidhara H S

Course Content

UNIT-I
Working with the Web - animations, concurrency, and working with the web, Guided project - Restaurant, a customizable menu app that displays the available dishes from a restaurant and allows the user to submit an order.

UNIT-II
Working with the Web - web service to set up the menu with their own menu items and photos, Improving Restaurant App with web services.

UNIT-III
Advanced Data Display - Collection Views: data source and delegate, compositional layout, diffable data sources, advanced compositional layout.

UNIT-IV
Swift: Generics types and protocols with associated types, generic functions, other UIKit topics: Search controllers, local notifications, Project: displaying a complex, dynamic data set with multiple layouts

UNIT-V
Prototyping and Project Planning - how to design, prototype, and architect a project.

Text Book:

1. Develop in Swift Data Collections, Apple Books

Course Outcomes (COs):

At the end of the course, students will be able to-
1. Apply animations and concurrency techniques to create an iOS App (PO-1,2,3,5,6,9,10,11,12) (PSO-)
2. Develop iOS applications to share data. (PO-1,2,3,5,6,9,10,11,12) (PSO-1)
3. Develop iOS applications using Collection Views (PO-1,2,3,5,6,9,10,11,12) (PSO-1)
4. Develop apps involving search controllers and local notifications (PO-1,2,3,5,6,9,10,11,12) (PSO-1)
5. Design the prototype of an iOS application. (PO-1,2,3,5,6,9,10,11,12) (PSO-1)

SYSTEM SOFTWARE

Course Code: ISE632   **Credit: 3:0:0**
Prerequisite: Nil   **Contact Hours: 42L**
Course Coordinator: Mrs. Pushpalatha M N

Course Content

UNIT-I
SIC,SIC/XE Architecture and Assembler: Introduction, System Software and Machine Architecture, Simplified Instructional Computer (SIC) – SIC and SIC/XE Machine Architecture, SIC Programming Examples, 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 (COs):

At the end of the course, students will be able to-
1. Understand the architecture of SIC, SIC/XE machine and instructions. (PO-1,2) (PSO-2)
2. Apply the assembler algorithm to generate the object code for SIC/XE (PO-1,2,3) (PSO-2)
3. Understand Loader functions, features and design options. (PO-1) (PSO-2)
4. Illustrate the concepts of macro processing and its various features. (PO-1) (PSO-2)
5. Design solutions using LEX and YACC for the given problem. (PO-1,2,3) (PSO-2)

SYSTEM SIMULATION AND MODELING

Course Code: ISE633   **Credit: 3:0:0**
Prerequisite: Engineering Mathematics   **Contact Hours: 42L**
Course Coordinator: Mrs. Vathsala M K

Course Content:

UNIT-I
Introduction to Simulation: When simulation is the appropriate tool and when it is not appropriate, Advantages and disadvantages of Simulation, Areas of application, Systems and system environment, Components of a system, Discrete and continuous systems, Model of a system, Types of Models, Discrete-Event System Simulation, Steps in a Simulation Study; **Simulation examples**: Simulation of queuing systems, Simulation of inventory systems

UNIT-II
Concepts in Discrete-Event Simulation: The Event-Scheduling / Time-Advance Algorithm, World Views, Manual simulation Using Event Scheduling; List processing, Simulation in Java, Simulation in GPSS; **Statistical Models in Simulation:** Review of terminology and concepts, Discrete distributions, Continuous distributions-Uniform distribution, Exponential distribution, Normal distribution

UNIT-III
Random-Number Generation: Properties of random numbers, Generation of pseudo-random numbers; Techniques for generating random numbers; Tests for Random Numbers. **Random-Variate Generation**: Inverse transform technique-Exponential Distribution, Uniform Distribution, Discrete Distributions, **Acceptance-Rejection technique**: Poisson Distribution, Convolution method

UNIT-IV
Queuing Models: Characteristics of queuing systems, Queuing notation, Long-run measures of performance of queuing systems; **Input Modeling**: Data Collection, Identifying the distribution with data, Parameter estimation, Goodness of Fit Tests, Selecting input models without data

UNIT-V
Verification and Validation of Simulation Models: Model building, verification and validation, Verification of simulation models, Calibration and validation of models, **Estimation of Absolute Performance**: Types of simulations with respect to output analysis, Stochastic nature of output data; Absolute measures of performance and their estimation

Text Book:

1. Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol: Discrete-Event System Simulation, Fifth Edition, Pearson Education, 2013.

References:

1. Lawrence M. Leemis, Stephen K. Park: Discrete – Event Simulation: A First Course, Pearson / Prentice-Hall, 2006.
2. Sheldon M. Ross: Simulation, Fourth Edition, Elsevier, 2006.
3. Averill M. Law: Simulation Modeling and Analysis, Fourth Edition, Tata McGraw-Hill, 2007

Course Outcomes (COs):

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

1. Understand the concepts used to develop simulation models. (PO-1,2) (PSO-2)
2. Apply discrete event and statistical models simulation techniques to solve the given problem. (PO-1,2,3) (PSO-2)
3. Apply various techniques for random number and random variate generation. (PO-1,2,3) (PSO-2)
4. Analyze Queueing and Input modeling techniques.   (PO-1,2) (PSO-2)
5. Understand the concepts of verification, validation and estimation of simulation models. (PO-1) (PSO-2)

BLOCKCHAIN ESSENTIALS AND DAPPS

Course Code: ISE634   **Credit: 3:0:0**
Prerequisite: Fundamentals of Distributed System   **Contact Hours: 42L**
Course Coordinator: Dr. Sanjay H A

Course Content:

UNIT-I

Distributed systems, CAP theorem, Byzantine Generals problem, Consensus. The history of blockchain, Introduction to blockchain, Various technical definitions of blockchains, Generic elements of a blockchain, Features of a blockchain, Applications of blockchain technology, Tiers of blockchain technology, Consensus in blockchain, CAP theorem and blockchain, Benefits and limitations of blockchain

UNIT-II

Decentralization using blockchain, Methods of decentralization, Blockchain and full ecosystem decentralization, Smart contract, Decentralized organizations, Decentralized autonomous organizations, Decentralized autonomous corporations, Decentralized autonomous societies Decentralized applications, Platforms for decentralization,
Cryptographic primitives : Symmetric cryptography , Asymmetric cryptography , Public and private keys Hash functions: Compression of arbitrary messages into fixed length digest, Easy to compute, Pre-image resistance, Second pre-image resistance, Collision resistance,Message Digest (MD),Secure Hash Algorithms (SHAs), Merkle trees, Patricia trees, Distributed hash tables (DHTs), Digital signatures, Elliptic Curve Digital signature algorithm (ECDSA)

UNIT-III

Bitcoin, Bitcoin definition, Transactions, The transaction life cycle, The transaction structure, Types of transaction, The structure of a block , The structure of a block header, The genesis block, The bitcoin network, Wallets, Smart Contracts-History, Definition, Ricardian contracts, Smart contract templates, Oracles, Smart Oracles, Deploying smart contracts on a blockchain, The DAO

UNIT-IV

Ethereum 101, Introduction, Ethereum clients and releases, The Ethereum stack, Ethereum blockchain, Currency (ETH and ETC), Forks, Gas, The consensus mechanism, The world state, Transactions, Contract creation transaction, Message call transaction, Elements of the Ethereum blockchain , Ethereum virtual machine (EVM), Accounts, Block, Ether, Messages, Mining, The Ethereum network Hands-on: Clients and wallets -Geth

UNIT-V

Hyperledger, Hyperledger as a protocol, Fabric, Hyperledger Fabric, Sawtooth lake, Corda

Text Book:

1. Imran Bashir. "Mastring BlockChain", Packt.

References:

1. Mastering Bitcoin: Programming the Open Blockchain Paperback – 2017 by Andreas M. O'rielly.

Course Outcomes (COs):

At the end of the course, students will be able to-
1. Illustrate the Blockchain terminologies with its applications. (PO-1) (PSO-2)
2. Analyse the working principles of Blockchain. (PO-1,2)   (PSO-2)
3. Comprehend the principles & methodologies used in Bitcoin and able to deploy smart contract (PO-1,2,3) (PSO-2)
4. Create Ethereum Network, Wallets, Nodes, Smart contract & Dapps (PO-1,2,3,5,7,9,10,12) (PSO-2)
5. Develop Block chain Based Application Architecture using Hyper ledger (PO-1,2,3,5,7,9,10,12)   (PSO-2)

CLOUD COMPUTING

Course Code: ISE641   **Credit: 3:0:0**
Prerequisite: Nil   **Contact Hours: 42L**
Course Coordinator: Dr. Siddesh G M

Course Content

UNIT-I

Introduction: Network centric computing and network centric content, Peer-to-peer systems, Cloud Computing, Cloud Computing delivery models & Services, Ethical issues, Cloud vulnerabilities, Challenges. **Cloud Infrastructure:** Amazon, Google, Azure & online services, open source private clouds. Storage diversity and vendor lock-in, intercloud, Energy use & ecological impact of data centers, service level and compliance level agreement, Responsibility sharing, user experience, Software licensing.

UNIT-II

Cloud Computing: Applications & Paradigms, Challenges, existing and new application opportunities, Architectural styles of cloud applications, Workflows: Coordination of multiple activities, Coordination based on a state machine model – the ZooKeeper, The MapReduce programming model, A case study: the GrepTheWeb application, Clouds for science and engineering, High performance computing on a cloud, cloud computing for biological research, Social computing, digital content, and cloud computing.

UNIT-III

Cloud Resource Virtualization: Virtualization, Layering and virtualization, Virtual machine monitors, Virtual machines, Performance and security isolation, Full virtualization and paravirtualization, Hardware support for virtualization, Case study: *Xen* -a VMM based on paravirtualization, Optimization of network virtualization in *Xen* 2.0, *vBlades* -paravirtualization targeting a *x86-64* Itanium processor, A performance comparison of virtual machines, The darker side of virtualization, Software fault isolation.

UNIT-IV

Cloud Resource Management and Scheduling: Policies and mechanisms for resource management, Applications of control theory to task scheduling on a cloud, Stability of a two-level resource allocation architecture, Feedback control based on dynamic thresholds, Coordination of specialized autonomic performance managers, A utility-based model for cloud-based web services, Resource bundling, combinatorial auctions for cloud resources, Scheduling algorithms for computing clouds, fair queuing, Start time fair queuing, Cloud scheduling subject to deadlines, Scheduling mapreduce applications subject to deadlines.

UNIT-V

Storage systems: Storage models, file systems, databases, DFS, General parallel File system, GFS, Apache Hadoop, Locks & Chubby, TPS & NOSQL databases, Bigdata, Mega store. **Cloud security:** Risks, Security, privacy and privacy impacts assessments, Trust, VM Security, Security of virtualization, Security risks in shared images.

Text Book:

1. Dan Marinescu, Cloud Computing: Theory and Practice, 1st edition, MK Publishers, 2013.

References:

1. Kai Hwang, Jack Dongarra, Geoffrey Fox, Distributed and Cloud Computing, From Parallel Processing to the Internet of Things, 1st edition, MK Publishers, 2012.
2. Anthony T. Velte, Toby J. Velete, Robert Elsenpeter, Cloud Computing: A Practical Approach, Tata McGraw Hill, 2010.

Course Outcomes (COs):

At the end of the course, students will be able to -
1. Apply the concepts of cloud delivery models and services.
   (PO-1,2,3,5,7,9,10,12) (PSO-2,3)
2. Build various cloud based applications. (PO-1,2,3,5,7,9,10,12) (PSO-2,3)
3. Illustrate different cloud resource virtualization strategies with case studies.
   (PO-1,7) (PSO-2,3)
4. Describe cloud resource management and scheduling policies (PO-1,7) (PSO-2,3)
5. Create cloud instances by applying storage models and security aspects.
   (PO-1,2,3,5,7,9,10,12) (PSO-2,3)

MOBILE COMPUTING

Course Code: ISE642   **Credit: 3:0:0**
Prerequisite: Nil   **Contact Hours: 42L**
Course Coordinator: Dr. B P Vijaya Kumar

Course Content

UNIT-I
Introduction: Challenges in mobile computing, coping with uncertainties, resource poorness, bandwidth, etc. Cellular architecture, co-channel interference, frequency, reuse, capacity increase bu cell splitting. **Evolution of mobile system**: CDMA, FDMA, TDMA, GSM. Wireless LAN: IEEE 802.11.

UNIT-II
Mobility Management: Cellular architecture, Co-channel interference, Mobility: handoff, types of handoffs; location management, HLR-VLR scheme, Mobile IP, Dynamic host configuration protocol, Mobile transport layer-Traditional and classical TCP.

UNIT-III
Databases: Database Hoarding Techniques, Data Caching, Transactional Models, Query Processing. **Data Dissemination and Broadcasting Systems:** Communication Asymmetry, Classification of Data-Delivery Mechanisms, Data Dissemination Broadcast Models, Selective Tuning and Indexing Techniques.

UNIT-IV
Data Synchronization in Mobile Computing Systems: Synchronization, Synchronization software for mobile devices, Synchronization protocols, SyncML - Synchronization language for mobile computing, Sync4J (Funambol), Synchronized Multimedia Markup Language (SMIL). Mobile Devices: **Server and Management**: Mobile agent, Application server, Gateways, Portals, Service Discovery, Device management, Mobile file systems, security.

UNIT-V
Support for Mobility- File Systems, Mobile operating systems; Features, services and interfacing modules of: Windows, Android, iOS for Mobile devices.

Text Books:

2. Rajkamal, Mobile Computing, Oxford University Press, 2nd Edition, 2012
2. Jochen Schiller, Mobile Communications, 2nd edition, Pearson, 2003.

Reference:

1. Reza B, Mobile Computing Principles, Cambridge University Press, 2005
2. Vijaya kumar B P, & Venkataram, P. Prediction-based location management using multilayer neural networks. Journal of Indian institute of science, 82(1), 7-22, 2002.
3. Vijaya kumar B. P, & Venkataram P. A neural network-based connectivity management for mobile computing environment. International Journal of Wireless Information Networks, 10(2), 63-71, 2003.

Course Outcomes (COs):

At the end of the course, students will be able to-
1. Describe the principles techniques and some of the analytics in mobile networks. (PO-1) (PSO-1,2)
2. Illustrate the concept of mobility and resource sharing in network and transport. (PO-1) (PSO-1,2)
3. Analyze the database handling, data dissemination, synchronization with respect to different Mobile OS. (PO-1,2,9,10) (PSO-1,2)
4. Describe the mobility support using different file systems and platforms. (PO-1,9,10) (PSO-1,2)
5. Develop mobile applications using different operating systems. (PO-1,2,3,5,9,10,11,12) (PSO-1,2)

SOFTWARE TESTING

Course Code: ISE643   **Credit: 3:0:0**
Prerequisite: Nil   **Contact Hours: 42L**
Course Coordinator: Dr. Naresh E

Course Content

UNIT-I

Review of Software Engineering: Software process models, Software engineering ethics, Software engineering challenges. **Requirements Analysis:** Requirements elicitation techniques, Functional and Non-functional requirements. **Software Design:** Architectural design and its styles, Object-oriented design. Implementation Issues.

UNIT-II

Perspective on Testing: Basic definitions, Test Scenarios, Test cases, Insights from a Venn diagram, identifying test cases, Error, fault and Failure taxonomies, Levels of testing, Activities of Test engineer, Test/Debug life cycle, testing principles, Testing throughout the SDLC. Examples: Generalized pseudocode, the triangle problem, The NextDate function, the commission problem, The SATM (Simple Automatic Teller Machine) problem, the currency converter.

UNIT-III

Functional Testing: Boundary value analysis, Robustness testing, Worst-case testing, Special value testing, Examples, Random testing, Equivalence classes, Equivalence test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations. Decision tables, Test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations.

UNIT-IV

Static Testing: Reviews, Types of reviews, Inspections, Inspection process, Inspection roles, benefits of inspection, Walkthroughs, Checklists. **Structural Testing:** Statement coverage testing, Condition coverage testing, Path coverage, computing cyclomatic complexity, exploratory testing.

UNIT-V

Test Management and Automation: Introduction, Test Planning, Test Reporting, Test Plan template. Test Automation, Terms used in Automation, Skills needed for automation, scope of automation, design and Architecture for automation, Process model for automation. **Test Metrics and Measurements:** Need and types of metrics.

Text Books:

1. Paul C. Jorgensen, Software Testing, A Craftsman"s Approach, 4th Edition, Auerbach Publications, 2017.
2. Graham Bath, Judy McKay, The Software Test Engineer's Handbook, 2nd Edition, Rocky Nook publisher, 2014.

3. Srinivasan Desikan, Gopalaswamy Ramesh: Software testing Principles and Practices, 2nd Edition, Pearson, 2007.
4. Ian Sommerville, Software Engineering, 9th Edition, Pearson Education, 2011.

References:

1. Andreas Spillner, Tilo Linz, Hans Schaefer: Software Testing Foundations, 2nd Edition, Shroff Publishers & Distributers Pvt ltd.
2. Rahul Shende, Testing in 30+ Open Source Tools, Shroff Publishers & Distributers Pvt ltd. 2010.
3. Aditya P Mathur, Foundations of Software Testing, Pearson, 2008.

Course Outcomes (COs):

At the end of the course, students will be able to-
1. Illustrate the knowledge of software engineering concepts. (PO-1) (PSO-1,2)
2. Understand the terms & concepts of software testing. (PO-1) (PSO-1,2)
3. Apply the concepts of validation and its techniques like black box testing and white box testing for a given problem. (PO-1,2,3,5,9,10,12) (PSO-1,2)
4. Analyze the concepts of verification and its techniques in the development of software. (PO-1,2,5,9,10,12) (PSO-1,2)
5. Design Test Scenarios and Test Cases with the reports to track and monitor the defects. (PO-1,2,3,5,9,10,12) (PSO-1,2)

ARTIFICIAL INTELLIGENCE

Course Code: ISE644   **Credit: 3:0:0**
Prerequisite: **Contact Hours: 42L**
Course Coordinator: Dr. Lincy Meera Mathews

Course Content:

UNIT-I

Introduction: What is AI? Foundation and History of Artificial Intelligence. Intelligent Agents: Agents and Environments, Rationality, The Nature of Environments, The Structure of Agents.

UNIT-II

Problem-solving by search: Problem Solving Agents, Example Problems, Searching for Solution, Uniformed Search Strategies, Informed Search Strategies, Heuristic Functions.
Adversarial Search: Games, Optimal Decisions in Games, Alpha Beta Pruning, Imperfect Real-Time Decision.

UNIT-III

Logical Agents: Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic : A very simple Logic, Agents Based on Propositional Logic, First Order Logic: Representation Revisited, Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic.

UNIT-IV

Interference in First-order Logic: Propositional vs. First-Order Inference, Unification and Lifting, forward chaining – First order Definite clauses, A simple forward chaining algorithm Backward chaining-A Backward Chaining Algorithm. Resolution - . Planning: Definition, Planning with State-Space Search, Planning Graphs.

UNIT-V

Uncertainty: Acting under Uncertainty, Basic Probability Notations, Inference using Full Joint Distributions, Independence, Bayes' Rule and its Use.
Learning: Reinforcement Learning, Introduction, Passive Reinforcement Learning, Active Reinforcement Learning.

Text Book:

1. Stuart Russel, Peter Norvig: Artificial Intelligence - A Modern Approach, 3rd Edition, Pearson Education, 2012.

2. Elaine Rich, Kevin Knight, Shivashankar B Nair: Artificial Intelligence, 3rd Edition, Tata McGraw Hill, 2011.

References:

1. Peter Jackson, "Introduction to Expert Systems", 3rd Edition, Pearson Education, 2007.
2. Deepak Khemani "Artificial Intelligence", Tata Mc Graw Hill Education 2013.
3. http://nptel.ac.in

Course Outcomes (COs):

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

1. Identify the fundamental characteristics and challenging issues of Artificial Intelligence (AI) systems (PO-1,2,3,4,12,) (PSO-2)
2. Apply various general purpose search algorithm as solutions for various problem-solving agents (PO-1,2,3,4,12,) (PSO-2)
3. Apply various symbolic knowledge representation to specify domains and reasoning tasks of a situated intelligent agent. (PO-4,5,9,12,) (PSO-2)
4. Apply algorithmic approach for planning and solving AI solutions that require problem solving, inference, perception, knowledge representation, and learning. (PO-1,4,5,6,7) (PSO-2)
5. Extract conclusions on learning and quantify the uncertainty in the conclusions obtained from uncertain knowledge. (PO-5,9) (PSO-2)

OBJECT ORIENTED PROGRAMMING WITH JAVA

Course Code: ISOE08   **Credit: 3:0:0**
Prerequisite: Fundamentals of Computing   **Contact Hours: 42L**
Course Coordinator: Mr. Koushik S

Course Content:

UNIT-I

Control Statements: Java's Selection Statements, if, switch, Iteration Statements, while, do-while, for, the For-Each Version of the for Loop, Nested Loops, Jump Statements, Using break, Using continue. **Introducing Classes:** Class Fundamentals, Declaring Objects, A Closer Look at new, Assigning Object Reference Variables, Introducing Methods, Constructors, Parameterized Constructors, The this Keyword, Instance Variable Hiding, Garbage Collection, The finalize Method.

UNIT-II

A Closer Look at Methods and Classes: Overloading Methods, Overloading Constructors, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final, Introducing Nested and Inner Classes, Exploring the String Class, Using Command-Line Arguments. **Inheritance:** Inheritance Basics, Using super, Creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding.

UNIT-III

Inheritance: Dynamic Method Dispatch, Why Overridden Methods? Using Abstract Classes, Using final with Inheritance, The Object Class. Packages and Interfaces: Packages, Access Protection, Importing Packages, Interfaces, Defining an Interfaces, Default Interface Methods, Use static Methods in an Interface, Final Thoughts on Packages and Interfaces

UNIT-IV

Exception Handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, multiple catch Clauses, Nested try Statements, throw, throws, finally. Exception Handling: Java's Built-in exceptions, Creating Your Own Exception Subclasses, Chained Exceptions.

UNIT-V

Type Wrappers: Character, Boolean, Numeric type wrappers. Autoboxing: Autoboxing and Methods, Autoboxing / Unboxing occur in expressions, Autoboxing/Unboxing Boolean and Character values, Autoboxing / Unboxing helps prevents errors. **The Collections Framework:** Collections Overview, The Collection Interfaces: the collection interface, the List interface, the Set interface. The Collection Classes: The ArrayList Class, The LinkedList Class.

Text Books:

1. Herbert Schildt, "Java: The Complete Reference", 9th Edition, McGraw Hill

Reference:

1. E. Balagurusamy; Programming with Java, McGraw-Hill; Sixth edition.

Course Outcomes (COs):

At the end of the course, students will be able to-
1. Design the class using java specific constructs to solve the given problems. (PO-1,2,3,5,9,10) (PSO-1)
2. Design solutions to real-world problems using UML concepts & diagrams. (PO-1,2,3) (PSO-1)
3. Develop solutions for dynamic programming by adopting the concepts of inheritance, packages and interfaces. (PO-1,2,3,5,9,10) (PSO-1)
4. Apply the concepts of Exception Handling to solve a given problem. (PO-1,3,5,9,10) (PSO-1)
5. Use the Collection framework to perform data manipulation operations. (PO-1,2,3,5,9,10) (PSO-1)

SOFTWARE TESTING

Course Code: ISOE09
Prerequisite: Nil
Course Coordinator: Dr. Naresh E

Credit: 3:0:0
Contact Hours: 42L

Course Content

UNIT-I
Review of Software Engineering: Software process models, Software engineering ethics, Software engineering challenges. **Requirements Analysis:** Requirements elicitation techniques, Functional and Non-functional requirements. **Software Design:** Architectural design and its styles, Object-oriented design. Implementation Issues.

UNIT-II
Perspective on Testing: Basic definitions, Test Scenarios, Test cases, Insights from a Venn diagram, identifying test cases, Error, fault and Failure taxonomies, Levels of testing, Activities of Test engineer, Test/Debug life cycle, testing principles, Testing throughout the SDLC. Examples: Generalized pseudocode, the triangle problem, The NextDate function, the commission problem, The SATM (Simple Automatic Teller Machine) problem, the currency converter.

UNIT-III
Functional Testing: Boundary value analysis, Robustness testing, Worst-case testing, Special value testing, Examples, Random testing, Equivalence classes, Equivalence test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations. Decision tables, Test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations.

UNIT-IV
Static Testing: Reviews, Types of reviews, Inspections, Inspection process, Inspection roles, benefits of inspection, Walkthroughs, Checklists. **Structural Testing:** Statement coverage testing, Condition coverage testing, Path coverage, computing cyclomatic complexity, exploratory testing.

UNIT-V
Test Management and Automation: Introduction, Test Planning, Test Reporting, Test Plan template. Test Automation, Terms used in Automation, Skills needed for automation, scope of automation, design and Architecture for automation, Process model for automation. **Test Metrics and Measurements:** Need and types of metrics.

Text Books:

5. Paul C. Jorgensen, Software Testing, A Craftsman's Approach, 4th Edition, Auerbach Publications, 2017.
6. Graham Bath, Judy McKay, The Software Test Engineer's Handbook, 2nd Edition, Rocky Nook publisher, 2014.

7. Srinivasan Desikan, Gopalaswamy Ramesh: Software testing Principles and Practices, 2nd Edition, Pearson, 2007.
8. Ian Sommerville, Software Engineering, 9th Edition, Pearson Education, 2011.

References:

4. Andreas Spillner, Tilo Linz, Hans Schaefer: Software Testing Foundations, 2nd Edition, Shroff Publishers & Distributers Pvt ltd.
5. Rahul Shende, Testing in 30+ Open Source Tools, Shroff Publishers & Distributers Pvt ltd. 2010.
6. Aditya P Mathur, Foundations of Software Testing, Pearson, 2008.

Course Outcomes (COs):

At the end of the course, students will be able to-
1. Illustrate the knowledge of software engineering concepts. (PO-1) (PSO-1,2)
2. Understand the terms & concepts of software testing (PO-1) (PSO-1,2)
3. Apply the concepts of validation and its techniques like black box testing and white box testing for a given problem. (PO-1,2,3,5,9,10,12) (PSO-1,2)
4. Analyze the concepts of verification and its techniques in the development of software. (PO-1,2,5,9,10,12) (PSO-1,2)
5. Design Test Scenarios and Test Cases with the reports to track and monitor the defects. (PO-1,2,3,5,9,10,12) (PSO-1,2)

DATA SCIENCE WITH PYTHON

Course Code: ISOE10 **Credit: 3:0:0**

Prerequisite: Nil **Contact Hours: 42L**

Course Coordinator: Dr. Pushpalatha M N

Course Content

UNIT-I

Introduction to Data Science: The data science process- The roles in a data science project, Stages of a data science project and setting expectations, the data science process and A Data Scientist's Role in this Process. **Data Visualization-** Basics of simple plotting, Line Chart vs Line Graph, Bar Graph, Pie Chart, Histogram, Frequency Polygons, Box Plot, Scatter Plot, Saving Plots or Graph or chart to a file.

UNIT-II

The way of the program: The Python programming language, what is a program?, What is debugging, Formal and natural languages, the first program, Debugging. Variables, expressions and statements, Functions, Conditionals **and recursion:** Modulus operator, Boolean expressions, Logical operators, Conditional execution, Alternative execution, chained conditionals, Nested conditionals, Recursion, Stack diagrams for recursive functions, Infinite recursion, Keyboard input, Fruitful functions: Return values, Incremental development, Composition, Boolean functions. Iteration: Multiple assignment, Updating variables, the while statement, break, Square roots, Algorithms.

UNIT-III

A string is a sequence: len, Traversal with a for loop, String slices, Strings are immutable, Searching, Looping and counting, String methods, the in operator, String comparison. Lists: A list is a sequence, Lists are mutable, traversing a list, List operations, List slices, List methods, Map, filter and reduce, deleting elements, Lists and strings, Objects and values, Aliasing, List arguments.

UNIT-IV

Dictionaries: Dictionary as a set of counters, Looping and dictionaries, Reverse lookup, Dictionaries and lists, Memos, Global variables, Long integers, Debugging. Tuples: Tuples are immutable, Tuple assignment, Tuples as return values, Variable-length argument tuples, Lists and tuples, Dictionaries and tuples, Comparing tuples, Sequences of sequences

UNIT-V

Python for Data Science: Managing data- Cleaning data and sampling for modeling and validation, Scikit-learn, Essential libraries and tools-Jupyter notebook, Numpy, Scipy, matplotlib, pandas, mglearn , Supervised Machine learning algorithms-K-Nearest Neighbor, A first Application-Classifying Iris Species.

Text Books:

1. Allen Downey, Think Python (How to Think Like a Computer Scientist), 2nd Edition by O'Reilly Media

2. Introduction to Machine Learning with Python, A guide for Data Scientists, Andreas C. Müller & Sarah Guido, O'Reilly Publications

3. Practical Data Science with R, Nina Zumel, John Mount, Manning Shelter Island.
4. https://python4csip.com/files/download/Data%20Visualization.pdf

References:

1. Learning Python, Fourth Edition, Mark Lutz, O'Reilly publication
2. Doing Data Science Cathy O'Neil and Rachel Schutt Straight Talk from The Frontline.O'Reilly 2014

Course Outcomes (COs):

At the end of the course, students will be able to-
1. Identify Data science problem for different domains (PO-2) (PSO-3)
2. Describe the basic programming concepts of python and Functions (PO-1) (PSO-3)
3. Apply appropriate data types/ data structures for the given problem using Lists, Dictionaries, Tuples. (PO-1,2,3,5,9,10,12) (PSO-3)
4. Apply python to solve different data science problems and visualize the data. (PO-1,2,3,5,9,10,12) (PSO-3)