Bharati Vidyapeeth

(Deemed to be University)

Pune, India

College of Engineering, Pune

B.Tech.(Computer Engineering) (2023 Course)

Program Curriculum

As Per NEP 2020 Guidelines

VISION OF UNIVERSITY:

Social Transformation through Dynamic Education

MISSION OF UNIVERSITY:

- To make available quality education in different areas of knowledge to the students as per their choice and inclination
- To offer education to the students in a conducive ambience created by enriched infrastructure! and academic facilities in its campuses.
- To bring education within the reach of rural, tribal and girl students by providing them substantive fee concessions and subsidized hostel and mess facilities
- To make available quality education to the students of rural, tribal and other deprived sections of the population

VISION OF THE INSTITUTE:

To be a World Class Institute for Social Transformation Through Dynamic Education.

MISSION OF THE INSTITUTE:

- To provide quality technical education with advanced equipment, qualified faculty members, and infrastructure to meet the needs of profession and society.
- To provide an environment conducive to innovation, creativity, research and entrepreneurial leadership.
- To practice and promote professional ethics, transparency and accountability for social community, economic and environmental conditions.

VISION OF THE DEPARTMENT

To pursue and excel in the endeavor for creating globally recognized Computer Engineers through quality education.

MISSION OF THE DEPARTMENT

- To impart engineering knowledge and skills confirming to a dynamic curriculum.
- To develop professional, entrepreneurial & research competencies encompassing continuous intellectual growth.
- To produce qualified graduates exhibiting societal and ethical responsibilities in working environment.

PROGRAM EDUCATIONAL OBJECTIVES

The students of B.TECH. (Computer Engineering), after graduating with Bachelor of Technology degree in Computer Engineering, will be able to

- Demonstrate technical and professional competencies by applying Engineering fundamentals, computing principles, and technologies.
- Learn, practice and grow as skilled professionals/entrepreneur/researchers adapting to the evolving computing landscape.
- Demonstrate professional attitude, ethics, understanding of social context and interpersonal skills leading to a successful career.

PROGRAM SPECIFIC OUTCOMES

- To design, develop and implement computer programs on hardware towards solving problems.
- To employ expertise and ethical practice through continuing intellectual growth and adapting to the working environment.

PROGRAM OUTCOMES

- Apply the knowledge of mathematics, science, engineering fundamentals, and computing for the solution of complex engineering problems.
- Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using computer engineering foundations, principles, and technologies.
- Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
- 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.
- Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
- Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal, and cultural issues, and the consequent responsibilities relevant to professional engineering practice.
- Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable development.
- Apply ethical principles while committed to professional responsibilities and norms of engineering practice.
- Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings

- Communicate effectively on complex engineering activities with the engineering community and with the 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.
- Apply the engineering and management principles to one's work, as a member and leader in a team.
- 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.

CORRELATION BETWEEN GRADUATE ATTRIBUTES AND PROGRAMME OUTCOMES

Graduate Attributes/ Programme Outcomes	a	b	c	d	e	f	g	h	i	j	k	l
Engineering Knowledge	✓											
Problem Analysis		✓										
Design/Development of Solutions			✓									
Conduct Investigations of Complex Problems				✓								
Modern Tool Usage					✓							
The Engineer and Society						✓						
Environment and Sustainability							✓					
Ethics								✓				
Individual and Teamwork									✓			
Communication										✓		
Project Management and Finance											✓	
Life-Long Learning												✓

- **DEFINITION OF CREDITS:**

1 Hour Lecture (L) per week	1 credit
1 Hour Tutorial (T) per week	1 credit
1 Hour Practical (P) per week	0.5 credits

- **STRUCTURE OF UNDERGRADUATE ENGINEERING PROGRAMME:**

Sr. No.	Category	Credit Distribution
1	Basic Science Courses	21
2	Engineering Science Course	28
3	Core Courses and Lab	90
4	Professional Elective Courses	12
5	Project	12
6	Internship	04
7	Skill Based Courses	10
*8	Value Added Courses	03 (Add on)
9	Humanity/Social Science Courses	03
10.	Massive Open Online Courses (MOOC)	02 (Add on)
11.	Social Activity	02 (Add on)
12.	Audit Course	02 (Add on)
TOTAL		**180**

- *** Indicates optional credits**

- **COURSE CODE AND DEFINITION**

Credits Per Semester

Sr. No.	Semester	Credits
1	I	25
2	II	25
3	III	20
4	IV	20
5	V	23
6	VI	22
7	VII	23
8	VIII	22
Total Credits		**180**

Course Code and Definition

Course Code	Definitions
L	Lecture
T	Tutorial
P	Practical
TW	Term Work
O	Oral
SEE	Semester End Examination
MJ	Major (Core) Courses
MI	Minor Courses

GE	General Elective Courses
OE	Open Elective Courses
SE	Skill Enhancement Courses
AE	Ability Enhancement Courses
VE	Vocational Enhancement Courses
VS	Vocational Skill Courses
VA	Value Added Courses
CC	Co-curricular Courses
EC	Extra-Curricular Courses
ID	Inter-disciplinary Courses
MD	Multidisciplinary Courses
RP	Research *I* Project Courses
PC	Practical Courses
BS	Basic Science
ES	Engineering Science
AC	Audit Course
EC	Extracurricular Activities
BM	Basic Mathematics
BP	Basic Physics
BC	Basic Chemistry
UH	Universal Human Values

BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY)
COLLEGE OF ENGINEERING, PUNE

Programme Code: 2311203 B. Tech. (Computer Engineering): Semester – III (2023 COURSE)

Sr. No	Category	Subject Code	Subject	Teaching Scheme			Examination Scheme-Marks						Credits			
				L	P	T	ESE	Internal Assessment	TW	PR	OR	Total	Th	Pr/Or	Tut	Total
1.	MJ	MJ1103301	Data Communication	3	-	-	60	40	-	-	-	100	3	0	0	3
2.	MJ	MJ1103302	Data Structures -II	3	2	-	60	40	25	25	-	150	3	1	0	4
3.	MJ	MJ1103303	Discrete Mathematics	3	2	1	60	40	25	-	25	150	3	1	1	5
4.	MJ	MJ1103304	Microprocessor and Microcontroller	3	2	-	60	40	25	-	25	150	3	1	0	4
5.	MJ	MJ1103305	System Programming	3	-	-	60	40	-	-	-	100	3	-	0	3
6.	SE	SE1103306	Java Programming	-	2	-	-	-	25	25	-	50	0	1	-	1
			Total	**15**	**8**	1	**300**	**200**	**100**	**50**	**50**	**700**	**15**	**4**	**1**	**20**
7.	AC	AC1113307	Indian Knowledge System*	2	-	-	-	100	-	-	-	100	-	-	-	2
8.	VA	VA1103308	VAC- I* i)**Introduction to Intellectual Property Rights** **ii) Introduction to Data Science**	2	-	-	-	100	-	-	-	100	-	-	-	2

*** Indicate this is mandatory course.**

AC: Audit Course VA : Value Added Course

Programme Code: 2311203 B. Tech. (Computer Engineering): Semester – IV (2023 COURSE)

Sr. No	Category	Subject Code	Subject	Teaching Scheme			Examination Scheme-Marks						Credits			
				L	P	T	ESE	Internal Assessment	TW	PR	OR	Total	Th	Pr/ Or	Tut	Total
1.	MJ	MJ1103401	Software Engineering	3	-	1	60	40	-	-	-	100	3	0	1	4
2.	MJ	MJ1103402	Operating System	3	-	-	60	40	-	-	-	100	3	-	0	3
3.	MJ	MJ1103403	Database Management System	3	2	-	60	40	25	25	-	150	3	1	0	4
4.	MJ	MJ1103404	Computer Network	3	2	-	60	40	25	-	25	150	3	1	0	4
5.	MJ	MJ1103405	Web Programming	3	2	-	60	40	25	25	-	150	3	1	0	4
6.	SE	SE1103406	Basic Python	-	2	-	-	-	25	25	-	50	0	1	-	1
			Total	**15**	**8**	-	**300**	**200**	**100**	**75**	**25**	**700**	**15**	**4**	**1**	**20**
7.	AE	AE1103407	MOOC – I*	-	-	-	-	-	-	-	-	-	-	-	-	2
8	EC	EC1103408	Social Activity*	-	-	-	-	-	-	-	-	-	-	-	-	2

*** Indicate this is mandatory course.**

Minor Course: Artificial Intelligent and Game Development

Sr. No.	Course Code	Name of Course	Teaching Scheme (Hrs./Week)			Examination Scheme (Marks)						Credits			
			L	P	T	ESE	IA	TW	OR	PR	Total	TH	PR/OR	Tut	Total
1	MI1103301	**Sem III** UI UX Design	3	2	-	60	40	25	25	-	150	3	1	-	**4**
2	MI1103401	**Sem IV** Game Theory	3	2	-	60	40	25	25	-	150	3	1	-	**4**
3	MI1103501	**Sem V** Knowledge Management System	3	2	-	60	40	25	25	-	150	3	1	-	**4**
4	MI1103601	**Sem VI** AI algorithms and Game Development	3	2	-	60	40	25	25	-	150	3	1	-	**4**
5	RP1103602	**Sem VI** Project	-	8	-	-	-	50	50	-	**100**	-	4	-	**4**
	Total		**12**	**16**	**-**	**240**	**160**	**150**	**150**	**-**	**700**	**12**	**8**	**-**	**20**

B. Tech (Computer Engineering)
Semester- III

Data Communication

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Practical:	--	Internal Assessment	40 Marks		
		Term Work	--	Practical	0
		Oral	--		
		Total	100 Marks	Total	03

Course Objective:

- To define different data formats for better data transmission.
- To introduce various digital baseband and bandpass modulation schemes.
- To identify the need of data coding and error detection/correction mechanism
- To provide knowledge of various multiplexing schemes.
- To get familiar with Cellular Telephony

Prerequisite: Digital electronics

Course Outcomes: On completion of the course, students will have the ability to:

1. Analyse data communication in TCP/IP protocol architecture
2. Classify types of data transmission and their characteristics
3. Demonstrate different types of Signal encoding techniques
4. Analyse transmission errors and correction techniques
5. Demonstrate bandwidth utilization using Multiplexing
6. Infer satellite networks used in cellular telephony

Unit I Data Communications **06 Hours**

Data Transmission and Network Capacity Requirements, Convergence, A Data Communication Model, The Transmission of Data Communication, Types of Networks, The Internet, Traditional Internet-Based Applications, Multimedia, Sockets Programming, The Need for a Protocol Architecture, The TCP/IP Protocol Architecture

Unit II Data Transmission **06 Hours**

Frequency, Spectrum and Bandwidth, Analog and Digital Data, Analog and Digital Signals, Analog and Digital Transmission, Asynchronous and Synchronous Transmission, Transmission Impairments -Attenuation, Delay and Distortion, Channel Capacity, Nyquist Bandwidth, Shannon Capacity, Noise, Guided Transmission Media, Wireless Transmission, Line-of-Sight Transmission

Unit III Signal encoding techniques **06 Hours**

Nonreturn to Zero (NRZ), Multilevel Binary Biphase Modulation Rate Scrambling Techniques, Amplitude Shift Keying, Frequency Shift Keying, Phase Shift Keying, Quadrature Amplitude Modulation, Pulse Code Modulation, Delta Modulation (DM), Angle Modulation

Unit IV Error detection and correction **06 Hours**

Types of Errors, Error Detection, Parity Check, the Internet Checksum, Cyclic Redundancy Check (CRC), Forward Error Correction, Forward Error-Correcting Codes, Flow Control, Error Control, High-Level Data Link Control (HDLC)

Unit V Multiplexing **06 Hours**

Frequency-Division Multiplexing, Analog Carrier Systems, Wavelength Division Multiplexing, Synchronous Time-Division Multiplexing, TDM Link Control, Digital Carrier Systems, SONET/SDH3 Cable Modems, Asymmetric Digital Subscriber Line, ADSL Design, Broadband Access Configuration, Multiple Channel Access, Frequency-Division Duplex (FDD)Time-Division Duplex (TDD)

Unit VI Cellular Telephony and Satellite Networks **06 Hours**

Cellular Telephony: Frequency reuse principle, Roaming, Handoff, Generations of cellular telephony, IS 95 standard, IMT 2000 Radio interface
Satellite Networks: Orbits and Footprints in satellite networks, Types of satellites, Trilateration in GPS
Case study- Iridium system, Global star system, Teleferic system

Textbooks

- Data and Computer Communications, Tenth Edition William Stallings, Pearson Education Limited
- Data communication and Networking, Fourth edition, Forouzan, MGH

Reference Books

- Leon W. Couch II, Digital and Analog Communication Systems, 6/E, Pearson Education Asia, 2002
- John J Proakis, Digital Communications, McGraw-Hill Higher Education, 2001
- Taub Schilling, Principals of Communication Systems, 2/E, Tata McGraw Hill, 2004

List of Project Based Learning Topics:

1. File transfer application
2. Remote desktop control
3. Design algorithms for bit-stuffing
4. Cyclic redundancy check calculator
5. Multiplexing and de-multiplexing of the signals
6. Polar & Bipolar Line Encodings Simulation
7. Design DM transmitter and receiver
8. Implementing a highly secure video steganography algorithm using Hamming code
9. Line Encoder and Scrambler with digital data generator and graph plotter using Python
10. Simulation of OFDM (Orthogonal Frequency Division Multiplexing) Signalling.

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Data Structures II

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Practical:	02 Hours/Week	Internal Assessment	40 Marks		
		Term Work	25 Marks	Practical	01
		Practical	25 Marks		
		Total	150 Marks	Total	04

Course Objective:
The course enables students to perform tasks that facilitate them to understand interaction. between the algorithms and the structure of the data being analyzed by these algorithms.

Prerequisite:
Classical Data Structure, Computational Thinking and Programming Concepts, Programming Technologies, and Tools Laboratory 3.

Course Outcomes: On completion of the course, students will have the ability to:

1. Identify and apply appropriate algorithms on the graph to solve real world problems.
2. Demonstrate the use of trees and binary search trees to solve the real-world problem.
3. Compare the different types of trees like AVL trees, B Tree, B+ Tree, Red Black tree and select an appropriate one to solve a particular one
4. Demonstrate use of hash function.
5. Identify and Use appropriate file organization for a real time application.
6. Explain the use of dictionaries and concept of text processing.

Unit I: Graph **06 Hours**

Introduction to Non-Linear data structure, Graphs, Representation of graph, AND/OR Graphs, ADT for Graph, Traversing a Graph, Dijkstra's Algorithm, Minimum Spanning Trees.

Unit II:Trees **06 Hours**

Introduction, Binary Trees, Binary Tree Representation, Tree Traversal Algorithms, Threaded Binary Tree, Binary Search Tree, Operations on Binary Search Tree, Huffman's Algorithm.

Unit III: Special forms of trees

AVL Trees, m-way Search Trees, B Trees, B+ Trees, Red Black Tree, 2-3 Trees, Splay Trees, Heaps and Binary Heaps . **06 Hours**

Unit IV : Hashing

Introduction, Hash functions, Collision Resolution Strategies, **06 Hours**
Types of Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing.

Unit V: Sorting and Indexing

Insertion Sort, Radix sort, Address Calculation Sort. Quick Sort, Merge Sort, Files and Their Organization: Data Hierarchy, File Attributes, Text Files and Binary Files, Basic File Operations, File Organizations, and indexing **06 Hours**

Unit VI : Dictionaries & Text Processing

Definition, Dictionary Abstract Data Type, Implementation of Dictionaries, Text Processing: String Operations, Brute-Force Pattern Matching, The Boyer-Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Standard tries, Suffix Tries, The Huffman Coding Algorithm **06 Hours**

Textbooks

- Data Structures: A Pseudo code approach with C, R. Gillberg, B. Forouzn
- Data structures using C and C++ by Langsam, Augenstein, Tenenbaum, PHI publication
- Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014.
- Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.

Reference Books

- Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, John Wiley & Sons, 2013
- Think Data Structures- Algorithms and Information Retrieval in Java, Allen B. Downey, O'Reilly, 2017

List of Laboratory Assignments - Provisional List of Assignments

1. Write a program to search for a particular number in a binary tree.
2. Write a program to display a mirror image of a binary search tree.
3. Write a program to perform the operations of insertion and deletion on the Heap.
4. Write a program to sort the given list of elements using Heap sort.
5. Create a graph and perform Depth First Search and Breadth First Search.
6. Create a record containing the cities and the distance between two cities. Find the optimal path between any two cities. The resulting path will contain all the cities taken into consideration.
7. Apply the following sorting techniques on the set of integers.
   8. Quick Sort.
   9. Merge Sort
10. Write a program to apply search using hashing technique on the student database of B.Tech Computer Engineering.

Project Based Learning - Provisional List of Projects

1. Student attendance system.
2. Car rental system.
3. Inventory management system
4. Design phone dictionary.
5. Dictionary using hierarchical data structure.
6. Quiz conduction application.

7. Subject recommendation system.
8. Hashing for cryptography
9. Payroll system
10. Network route identifier.
11. Path finder
12. Telephone directory
13. Library Management system
14. Document indexing
15. Data Compressor
16. Railway reservation system
17. Develop LR parser
18. Develop a lexical analyzer

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Discrete Mathematics

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Practical:	02 Hours/Week	Internal Assessment	40 Marks	Tutorial	01
Tutorial	01 Hours/Week	Term Work	25 Marks	Practical	01
		Oral	25 Marks		
		Total	150 Marks	Total	05

Course Prerequisites: - Basic knowledge of Mathematics, linear algebra, probability and statistics.

Course Objectives: - The course emphasis on mathematical foundations required for computing enabling the students to develop logical thinking, reasoning, and problem-solving skills.

Course Outcomes: On completion of the course, students will have the ability to:

1. Demonstrate the ability to write the sentences in the symbolic logic and use of appropriate proof techniques.
2. Apply the basic principles of set theory to solve real world problems.
3. Apply properties of relations and functions to solve real time problems.
4. Demonstrate use of algebraic structures to solve computational problems.
5. Solve problems of combinatorics and recurrence relations.
6. Demonstrate use of discrete structure in computer science applications.

Unit I Mathematical Logic **06 Hours**

Connectives, Normal forms, Rules of inference, Predicate Calculus Quantifiers, Universe of Discourse, Mathematical induction, Greatest Common Divisor, Euclidean algorithm, Congruence relation/equations, Chinese Remainder Theorem, Cryptography.

Unit II Set Theory **06 Hours**

Types of sets, Set operations and laws, Venn Diagrams, Algebra of Sets, Multisets, Counting Principle, Cardinality, Addition Principle, Principle of inclusion and exclusion and its application.
Partitions: Ordered and Unordered Partitions

Unit III Relations **06 Hours**

Relations: Cartesian Product, Basic definition, properties and types of relations, Digraphs, equivalence, and partial ordered relations, n-ary relations, Transitive closure and Warshall's algorithm.
Functions: Types of functions, Identity functions, Composition of functions, Inverse functions, Hash function, Recursive functions.

Unit IV **06 Hours**

Algebraic Structures Operations and its properties, Semigroups, Groups, Subgroups, Homomorphism Groups, Rings, Cyclic groups.

Lattice: Ordered sets, Posets and Hasse Diagrams, Supremum and Infimum, Properties of lattices. Generation of Codes, Parity Checks, Error recovery.

Unit V Counting and Recurrence Relations **06 Hours**

Permutations, Combinations, The Pigeonhole Principle, Recurrence Relation, Linear Recurrence Relations with constant Coefficients, Total solutions.

Unit VI Concepts of Graphs and Trees: Definition, Degree, Types, Operations on graphs, Paths, Circuits, Planar graphs and their properties, Eulerian and Hamiltonian graphs. **06 Hours**

Trees: Basic properties of trees, Binary trees, Application: Minimum Spanning Tree, Shortest Path, Huffman coding.

Textbooks

- J.P. Tremblay and Manohar: Discrete mathematical structures with application to Computer Science, McGraw hill- New Delhi.
- B. Kolman and R.C. Busby: Discrete mathematical structures for computer science Prentice Hall, New-Delhi.
- S. Malik and M. K. Sen Discrete Mathematics, Cengage Learning India Pvt. Ltd.

Reference Books

- Kenneth H. Rosen, Discrete Mathematics, and its applications Eighth Edition McGraw Hill Education
- Stanat and McAlister, Discrete Mathematics for Computer Science, PHI
- R.M. Somasundaram Discrete Mathematical Structures, Prentice Hall India Learning Private Limited

List of Laboratory Exercises

1. Given a fact or a statement prove or disprove using suitable technique.
2. Use the Euclidean algorithm to find the greatest common divisor (GCD) of two numbers.
3. Write the given English language sentences represent in the Symbolic logic
4. Given the statement forms Infer the validity of the statement form.
5. Perform set Operations and Compute a power set of a given set.
6. Apply Warshall's algorithm to compute a Transitive Closure of a given relation entered by the user. (Use any suitable programming language).
7. Draw a Hasse diagram and find chains and antichains.
8. Find the number of ways for any event.
9. Given a problem represent in a graph and compute the optimal solution.
10. Use Dijkstra's algorithm to find the shortest path in a weighted graph.
11. Apply Kruskal's algorithm to find the minimum spanning tree for any weighted graph.
12. Determine if the graph has an Eulerian path or circuit. If so, provide the path or circuit.

Project Based Learning - Provisional List of Projects

1.Discrete Mathematics in Railway Planning using graph theory and linear algebra.
2.Object transformations using linear algebra.
3.Discrete mathematics in cryptography.
4.In Google maps to determine fastest driving routes and times.

5.In image processing
6.In relation database using sets.
7.In cyber security using graph theory.
8.Shortest path between two cities using a transportation system.
9.Data compression system with the help of Huffman coding.
10.Find the shortest tour that visits each of a group of cities only once and then ends in the starting city using graphs.

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Microprocessor and Microcontroller

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Practical:	02 Hours/Week	Internal Assessment	40 Marks		
		Term Work	25 Marks	Practical	01
		Oral	25 Marks		
		Total	150 Marks	Total	04

Course Objective:

- To develop an understanding of the architecture and functions of microprocessors and micro-controllers.
- To learn machine language programming & interfacing techniques.

Prerequisite: Digital Electronics

Course Outcomes: On completion of the course, students will have the ability to:

1. Describe the architecture of 80386DX along with addressing modes.
2. Differentiate between real and protected mode and use programmer's model.
3. Describe interfacing of microprocessor with programmable peripheral devices.
4. Explain concepts of Multicore architecture.
5. Differentiate between microprocessor and microcontroller and describe core concepts related to PIC architecture.
6. Describe concepts related to display technologies and interfacing.

Unit I **06 Hours**

Introduction to 80386

Concepts of architecture of 80386 DX, Registers, Salient features of 80386 DX, Signal definition, addressing modes, Instruction format, Instruction pipelining in80386.

Unit II **06 Hours**

Real and protected mode

Real mode programming model, Memory addressing in real mode, Interrupt Handling, and exceptions, switching between real and protected mode, protected mode register model.

Unit III I/O interface **06 Hours**

Programmable peripheral devices and interfacing, Interfacing with Programmable Peripheral Interface8255, Programmable Interrupt Controller
8259, USART8251, Programmable Interval Timer8253, DMA Controller8237, Block diagram, operating modes and control words.

Unit IV Multicore Architecture **06 Hours**

Intel 64-bit architecture, Pentium processor functional block diagram, memory management, Multicore Architecture, Bus Connections, core to duo and dual core processors, characteristics, and design guidelines.

Unit V **06 Hours**

Introduction to PIC microcontroller

Comparison of microprocessor and microcontroller, Micro controllers and embedded processors, Overview of PIC Family, PIC Architecture, Registers, and memory in PIC, WREG register, File register, Status register Special Function Registers, RISC architecture in PIC, I/O ports

Unit VI Display Technologies **06 Hours**

LCD, LED and 7 Segment Interfacing Different peripheral device -Different types of display units -7 Segments & its types -Principle of Operation-Common Anode mode-Common Cathode mode -16x2 LCD - Applications-Hardware interfaces-Interfacing Circuits for LCD & LED -Pin diagram of 16x2- working mechanism LCD using Arrays & Pointers.

Textbooks

- Intel microprocessor and peripheral handbook (32bit)80386DX
- Muhammad Ali Mazidi"PIC Microcontroller and embedded systems",2009,Pearson education
- 80386Microprocessor Handbook,Chris H.Pappas,William H.Murray

Reference Books

- D.V. Hall,"Micro Processor and Interfacing",TataMcGraw-Hill.
- Intel 64 and IA-32-bit architectures Software Developer's Manual, Volume 3A, Intel, (Digital Content PDF: 253668.pdf)
- Microcomputer Systems:The8086/8088Family:Architecture,Programming and Design,2nd ed., Liu&Gibson

List of Assignments

1. Write an assembly language program to accept a string and display its length.
2. Write an assembly language program to perform non-overlapped block transfer without string specific instruction. Block containing data can be defined in data segment.
3. Write an assembly language program to find factorial of user entered number.
4. Write an assembly language program.
5. Write a simple PIC assembly program that uses the WREG register for temporary data storage.
6. Write a program to display numbers 0-9 on the 7-segment display.
7. Interface an LED to a PIC microcontroller.

List of Laboratory Exercises

1. Write an assembly language program for multiplication of two 32-bit numbers.
2. Write an assembly language program to count number of negative and positive numbers from given array.
3. Write an assembly language program to check whether entered string is palindrome or not.
4. Write an assembly language program to arrange given set of numbers in ascending order.
5. Write an assembly language program to arrange given set of numbers in descending order.
6. Write an assembly language program display current time from system

7. Write an assembly language program to detect protected mode and display values of GDTR, LDTR,IDTR,TR and MSW registers.
8. Interface stepper motor/DC motor with PIC microcontroller.
9. Interface LCD with PIC microcontroller display and display certain message.
10. Interface keyboard with PIC microcontroller.

Project Based Learning - Provisional List of Projects

1. Basic calculator
2. Ultrasonic Distance Meter with Buzzer indication
3. RFID based Attendance System
4. Temperature and Humidity Measurement
5. Digital Door Lock System
6. Gate Controller with High-Speed Alerting System
7. Celsius Scale Thermometer
8. Game Console

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

System Programming

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
		Internal Assessment	40 Marks		
		Total	100 Marks	Total	03

Course Objective: To provide students basic knowledge of applications programs, system programs, overview of language translation and operating system.

Prerequisite:
Basic computer system hardware, I/O devices

Course Outcomes: On completion of the course, students will have the ability to:

1. Interpret theoretical and practical aspects of language translation
2. Examine working and design of assemblers and microprocessors
3. Explain the concept of memory allocation, relocation and the functions of loaders, linkers
4. Examine the phases of compiler and its working.
5. Demonstrate the various operating system services.
6. Demonstrate use of device drivers

Unit I System Software and Assemblers: **06 Hours**

Definition, Components of system software, Evolution of system software, Language translators, Machine Structure, Machine Language, Assembly Language, High-level Language, Language translation basics,

Unit II: Assembler and Macro processor **06 Hours**

Structure of an assembler, Design of a Single Pass and Two Pass Assembler Macro language and macro processor, macro instructions, features of macro facility, macro instruction arguments, conditional macro expansion, macro call within macros, macros instructions defining macros, Microprocessor design

Unit III: Linkers and Loaders:

Loader scheme, absolute loaders, subroutine linkages, relocating loaders, direct linking loaders, other loader schemes Binders, linking loaders, Overlays, Dynamic Binders, Design of an absolute Loaders, Design of a Direct –Linking loaders Dynamic Link Libraries. **06 Hours**

Unit IV: Compiler

Basic Compiler Function Compiler phases -Lexical Analysis, Syntax Analysis: Role of Parser, Top-down parsing, Recursive dent and predictive parsers (LL), Bottom-Up parsing, Operator precedence **06 Hours**

parsing, LR, SLR and LALR parsers. Intermediate code generation three address code intermediate code forms, Compiler generation tools – LEX and YACC. Interpreters.

Unit V: Operating system

system concept, Operating system structure. O.S. Components, O.S. Services, System calls. Shell scripting Shell scripting (Bourne Shell (SH), Bourne-Again Shell (BASH), C-Shell (CSH), TCSH, Korn Shell (KSH)) Shell commands (Basics, Pipelining Background/Foreground, File Permissions, etc.) AWK Programming, Process control. **06 Hours**

Unit VI: Device Drivers

Definition, Anatomy and types, Device Programming, Installation and Incorporation of driver routines, Basic device driver operation, Implementation with Line printer, Comparative study between device drivers for Unix and Windows **06 Hours**

Textbooks

- System Programming by John J. Donovan, TATA McGraw-Hill Edition.
- System Programming by Leland Beck, Pearson Ed.
- D. M. Dhamdere : "Systems programming and operating system", Tata McGraw Hill

Reference Books

- Unix programming Environment- Keringham and Pike, Pearson Education

Project Based Learning - Provisional List of Projects

1. Develop an assembler for 8086(Single Pass)
2. Develop an assembler for 8086(Two Pass
3. Develop a Macroprocessor for 8086 (Single Pass)
4. Develop a Macroprocessor for 8086 (Two Pass)
5. Demonstrate phases of Compiler
6. Develop a LALR Parser
7. Develop LR Parser
8. Develop SLR parser and demonstrate it's use

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

SBC-III Java Programming

Teaching Scheme		Examination Scheme		Credit Scheme	
Hours/Week			Marks		Credits
		Term Work	25 Marks	Practical	01
Practical:	02 Hours/Week	Oral	25 Marks		
		Total	50 Marks	Total	01

Course Objective:

- To teach fundamentals of object-oriented concepts and Java programming.
- To apply the concepts of object-oriented paradigm.
- To develop Java programming skills.
- To design and implement applications for real life problems by using Java. programming.

Prerequisite:

Paradigms of Programming

Course Outcomes: On completion of the course, students will have the ability to:

1. Analyse the basics of object-oriented programming with Java programming
2. Identify class, objects, Methods for real time problems
3. Make use of constructor, Garbage Collector and methods of string class
4. Demonstrate the concept of inheritance and polymorphism with the help of real time applications
5. Infer the exception with exception handling mechanism and multi-threading programming.
6. Design the graphical user interface by using Applets and AWT, SWING.

Unit I: **06 Hours**

Introduction: History and Features of Java, Difference between Java , C, C++, Internals of Java Program, Difference between JDK, JRE and JVM, Internal Details of JVM,

Basics of Java Language-Variable and Reserve / Keywords present in Java, Primitive Data types, Java Operators, Decision making and branching statements in Java

Unit II: **06 Hours**

Classes, Objects and Methods:

Creating a Class, Visibility/Access Modifiers, Encapsulation, Methods: Adding a Method to Class, returning a Value, adding a Method That Takes Parameters, ‘this’ Keyword, Method Overloading, Object Creation, Using Object as a Parameters, Returning Object, Array of Objects, Memory Allocation: ‘new’, Static Data Members, Static Methods,

Unit III: **06 Hours**

Constructors, Destructors and String Handling:
Use of Constructor, Characteristics of Constructors, Types of Constructors, Constructor Overloading, Constructor with Default Arguments, Symbolic Constants, Garbage Collection, Destructors and Finalizers.
String Handling:
String: Immutable String, String Comparison, String Concatenation, Substring, Methods of String class, String Buffer class, StringBuilder class, Creating Immutable class, to String method

Unit IV: **06 Hours**

Use of Inheritance, Types of Inheritance in Java, Role of Constructors in inheritance, Polymorphism in OOP, Types of Polymorphism, static and dynamic polymorphism, Overriding Super Class Methods. Use of "super" keyword. Interfaces, Implementing interfaces.

Unit V: **06 Hours**

Exception Handling and Multithreaded programming:
Exception Handling: try and catch block, catch block, Nested try, finally block, throw keyword, Exception Propagation, throws keyword, Exception Handling with Method Overriding, Custom Exception
Introduction to threads, life cycle of a thread, thread states, thread properties, methods in Threads and Runnable, setting priority of threads, synchronization and inter thread communication Life Cycle of a Thread

Unit VI: **06 Hours**

Designing Graphical User Interfaces in Java
Applet and its use Design Patterns using Applet and JApplet. Run Applet application by browser and applet tool. Applet Architecture. Parameters to Applet Life Cycle of
Components and Containers Basics of Components Using Containers Layout Managers and user defined layout. Border Layout , Flow Layout , Grid Layout , GridbagLayout, BoxLayout. AWT Components Adding a Menu to Window Extending GUI Features Using SWING Components Designing GUI. Advanced swing components like JProgressbar, JSlider, JRadioButton , JTree, JTable, JToggleButton, etc.

Textbooks:

- E. Balaguruswamy, "Object Oriented Programming Using C++ and Java", Tata McGrawHill.
- Steven Holzner et al. "Java 2 Programming", Black Book, Dreamtech Press, 2009.

Reference Books

- Java The complete reference, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd. 9th edition, 2014, ISBN: 978-0-07-180856-9 (E-book)

- Object-Oriented Design Using Java, Dale Skrien, McGraw-Hill Publishing, 2008, ISBN - 0077423097, 9780077423094.
- Mitsunori Ogihara, “Fundamentals of Java Programming”, Springer; 2018, ISBN 978-3-319-89490-4.
- Brahma Dathan Sarnath Ramnath, “Object-Oriented Analysis, Design and Implementation An Integrated Approach”, Springer; 2nd ed. 2015, ISSN 1863-7310 ISSN 2197-1781 (electronic) Undergraduate Topics in Computer Science ISBN 978-3-319-24278-1, ISBN 978-3-319-24280-4.
- T. Budd (2009), An Introduction to Object Oriented Programming, 3rd edition, PearsonEducation, India.
- J. Nino, F. A. Hosch (2002), An Introduction to programming and OO design using Java, John Wiley & sons, New Jersey.
- Y. Daniel Liang (2010), Introduction to Java programming, 7th edition, Pearson education, India.

Laboratory Experiments:

1. Write a Java program that checks whether a given string is a palindrome or not.
2. Write a Java program that describes a class person. It should have instance variables to record name, age and salary. Create a person object. Set and display its instance variables.
3. Write a Java program that creates a class circle with instance variables for the centre and the radius. Initialize and display its variables.
4. Write a Java program that counts the number of objects created by using static variable.
5. Write a Java program to demonstrate the constructors in java.
6. Write a Java program to demonstrate the constructor overloading.
7. Write a Java program to display the use of this keyword.
8. Write a Java program to implement Class and Inheritance Concept.
9. Write an application that creates an interface’ and implements it.
10. Write a program that can count the number of instances created for the class.
11. Write a program to implement the concept of Multithreaded Programming.
12. Create an abstract class shape. Let rectangle and triangle inherit this shape class. Add necessary functions.
13. Write an application that shows the usage of try, catch, throws and finally.
14. Write an Applet that displays ―Hello World (Background color-black, text color-blue and your name in the status window.)

Indian Knowledge System

TEACHING SCHEME:	**EXAMINATION SCHEME:**	**CREDITS ALLOTTED:**
Theory: 02	End Semester Examination: --	Theory: 02
Practical: 00	Internal Assessment: 100 Marks	
Tutorial: 00		
		Total Credit: 02

Course Objectives:

1. To sensitize the students about Indian culture and civilization including its Knowledge System and Tradition.
2. To help student to understand the knowledge, art and creative practices, skills, and values in ancient Indian system
3. To help to study the enriched scientific Indian heritage.
4. To introduce the contribution from Ancient Indian system & tradition to modern science & Technology

Course Outcomes: After learning this course students will be able to understand

1 Concepts of Indian Knowledge System

2 India's contribution in Philosophy and Literature

3 India's involvement in Mathematics and Astronomy

4 India's role in Medicine and Yoga

5 India's influence in Sahitya

6 Concepts of Indian Shastra

UNIT – I **Introduction to Indian Knowledge System** **(04 Hours)**

Definition, Concept and Scope of IKS, IKS based approaches on Knowledge Paradigm, IKS in ancient India and in modern India

UNIT – II **Philosophy and Literature** **(04 Hours)**

Contributions by Maharishi Vyas, Manu, Kanad, Pingala, Parasar, Banabhatta, Nagarjuna and Panini in Philosophy and Literature

UNIT - III **Mathematics and Astronomy** **(04 Hours)**

Contribution of Aryabhatta, Mahaviracharya, Bodhayan, Bhashkaracharya, Varahamihira and Brahmgupta in Mathematics and Astrononmy

UNIT -IV **Medicine and Yoga** **(04 Hours)**

Major contributions of Charak, Susruta, Maharishi Patanjali and Dhanwantri in Medicine and Yoga

UNIT -V **Sahitya** **(04 Hours)**

Introduction to Vedas, Upvedas, Upavedas (Ayurveda, Dhanurveda, Gandharvaveda)
Puran and Upnishad) and shad darshan (Vedanta, Nyaya.Vaisheshik, Sankhya, Mimamsa,
Yoga, Adhyatma and Meditation)

UNIT -VI **Shastra** **(04 Hours)**

Introduction to Nyaya, vyakarana, Krishi, Shilp, Vastu, Natya and Sangeet

Reference Books

1.Textbook on IKS by Prof. B Mahadevan, IIM Bengaluru

2. Kapur K and Singh A.K (Eds) 2005). Indian Knowledge Systems, Vol. 1. Indian Institute of Advanced Study, Shimla. Tatvabodh of sankaracharya, Central chinmay mission trust, Bombay, 1995.

3. The Cultural Heritage of India. Vol.I. Kolkata:Ramakrishna Mission Publication, 1972.

4. Nair, Shantha N. Echoes of Ancient Indian Wisdom. New Delhi: Hindology Books, 2008.

5. Dr. R. C. Majumdar, H. C. Raychaudhuri and Kalikinkar Datta: An Advanced History of India (Second Edition) published by Macmillan & Co., Limited, London, 1953.

6. Rao, N. 1970. The Four Values in Indian Philosophy and Culture. Mysore: University of Mysore.

7. Avari, B. 2016. India: The Ancient Past: A History of the Indian Subcontinent from c. *7000* BCE to CE 1200. London: Routledge.

8. Textbook on The Knowledge System of Bhārata by Bhag Chand Chauhan,

9. Histrory of Science in India Volume-1, Part-I, Part-II, Volume VIII, by Sibaji Raha, et al. National Academy of Sciences, India and The Ramkrishan Mission Institute of Culture, Kolkata (2014).

10. Pride of India- A Glimpse of India's Scientific Heritage edited by Pradeep Kohle et al. Samskrit Bharati (2006).

12. Vedic Physics by Keshav Dev Verma, Motilal Banarsidass Publishers (2012).

13. India's Glorious Scientific Tradition by Suresh Soni, Ocean Books Pvt. Ltd. (2010).

14.Kapoor, Kapil, Avadesh Kr. Singh (eds.) *Indian Knowledge Systems* (Two Vols), IIAS, Shimla, 2005

Minor Course: Artificial Intelligence and Game Development

Semester-III

UI UX Design

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	**03 Hours/Week**	**University Examination**	**60 Marks**	**Lecture**	**03**
Practical:	**02 Hours/Week**	**Internal Assessment**	**40 Marks**		
Tutorial:	**00 Hours/Week**	**Term Work**	**25 Marks**	**Practical**	**01**
		Oral	**25 Marks**		
		Total	**150 Marks**	**Total**	**04**

Course Objective:

- Understand the definition, difference, and similarities of user experience and usability and apply these to design, research, and testing practices in field of interaction design for digital interfaces.

Prerequisite: Basic knowledge of designing tools and languages like HTML, Java, etc
-

Course Outcomes: On completion of the course, students will have the ability to:

1. Identify User Interface (UI) design principles.
2. Apply Interactive Design process in real world applications.
3. Judge user research, solution conceptualization and validation as interwoven activities in the design and development lifecycle.
4. Infer the ability to constructively engage with the Design professionals they would work with in the future.
5. Understand user-centered design and usability engineering principles as they design a variety of software user interfaces.
6. Use prototyping methods to discover requirements and to evaluate design alternatives.

Unit I: Introduction to UI: **06 Hours**

User interface design basic, relationship between UI and UX, Historical review of interface design, interface conventions, template vs contents, formal elements of interface design, UI design process.

Unit II: Introduction to UX **06 Hours**

UX basics, good and poor design, understanding your user, designing the experience, Elements of UX, Visual design principles, Functional layout, Introduction to interface.

Unit III: UX Design Process: **06 Hours**

Types of Data in UX Design, Introduction and Definition on 6 Stages of Design Process, Heuristic Evaluation, Information Design and Data Visualization, Wire framing & Storyboarding, Elements and Widgets,

Unit IV: UX Testing Process: **06 Hours**

Design Testing Methods and Techniques. Usability Testing – Types and Process, Various Prototyping Tools, How to prepare Usability Testing, How to understand & refine Usability Test Results?

Unit V: UI/ UX Design Tools: **06 Hours**

User Study- Interviews, writing personas: user and device personas, User Context, Building Low Fidelity Wireframe and High-Fidelity Polished Wireframe Using wireframing Tools, Creating the working Prototype using Prototyping tools, Sharing and Exporting Design

Unit VI: Cognitive Psychology in Design: **06 Hours**

Cognitive psychology principles guide design decisions, understanding how users perceive, process, and remember information informs design decisions, Hick's Law and Fitts' Law, Designing for short and long-term memory retention.

Textbooks

- The Elements of User Experience: User-Centered Design for the Web and Beyond, Second Edition Jesse James Garrett, Pearson Education
- he UX Book Process and Guidelines for Ensuring a Quality User Experience, Rex Hartson and Pardha S. Pyla, Elsevier,
-

Reference Books

- The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, Third Edition Wilbert O. Galitz , Wiley Publishing,

Laboratory Experiments:

1. Design a Logo for an E-Commerce app
2. Design an email that showcases a promotional offer of the e-commerce app
3. Create User Personas, User Stories, and Use Case Diagrams to de□ine a Problem Statement, Scope, and understand the problem.
4. User research: Document & Conduct user, market, product, and competitive research to find possible solutions.
5. Ideation: Create sketches and low-fidelity wireframes of the scoped solution.
6. Execution: Create High-Fidelity Mockups & Prototypes from the wireframes.

7. Figma basics: Creating basic responsive elements like buttons, input elements, etc. to understand frames, groups, layout, constraints, texts, vector, color palette.
8. Basic Clickable Prototyping using figma.
9. Create a Design System for an e-commerce app using Grid and Spacing, Typography, Color System, and UI elements like icons, images, buttons, Inputs, Cards, Search Bar, Lists, etc.

Project Based Learning - Provisional List of Projects

1. Project Web Site and Brainstorming.
2. UI Critique.
3. Study the process of creating Graphically User Interface.
4. Study the implementation of GUI to different devices.
5. Develop the complete design process.

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Value Added Course-I : Introduction to Data Science

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	02 hours/Week	University Examination	Marks	Lecture	02
Practical:		Internal Assessment	100 Marks		
		Term Work	-		
		Oral	-		
		Total	100 Marks	Total	02

Course Objective:

The objective of the course "Introduction to Data Science" is to provide students with foundational knowledge and practical skills in the field of data science. This course aims to empower students with the foundational knowledge, skills, and ethical awareness necessary to embark on further studies or pursue careers in data science and related fields.

Prerequisite:

Students should have a basic understanding of programming concepts. A solid foundation in mathematics, including algebra, calculus, and probability theory, is highly recommended. Students should have a basic understanding of data and its various forms (e.g., structured, unstructured, semi-structured). Familiarity with spreadsheet software (e.g., Microsoft Excel, Google Sheets) for data manipulation and analysis is advantageous.

Course Outcomes: On completion of the course, students will have the ability to:

1. Describe different data sources, collection methods, and storage techniques
2. Utilize R for data manipulation, analysis, and visualization.
3. Calculate and interpret descriptive statistics (mean, median, mode, variance, standard deviation).
4. Apply data preprocessing methods i
5. Explain basic machine learning concepts and algorithms.
6. Apply principles of effective data visualization

Unit I Introduction to Data Science: Definition of data science, Importance and applications of data science in various industries, Foundations of Data Science, Data types (structured, unstructured, semi-structured), Data sources and data collection methods, Data storage and retrieval techniques. **06 Hours**

Unit II: Programming for Data Science: Getting started with R and RStudio, R basics. **06 Hours**

Unit III : Statistical Inference: Populations and samples - Statistical modeling, probability distributions, fitting a model, Exploratory Data Analysis and the Data Science Process - Basic tools (plots, graphs and summary statistics) of EDA - Philosophy of EDA - The Data Science Process **06 Hours**

Unit IV : Data Cleaning and Preprocessing: Data cleaning techniques (handling missing values, outliers),Data preprocessing techniques (scaling, normalization, feature engineering) **06 Hours**

Unit V : Exploratory Data Analysis (EDA): Data summarization and visualization, Identifying patterns and trends in data, Correlation analysis **06 Hours**
Machine Learning Basics: Introduction to machine learning concepts and algorithms ,Supervised learning (classification, regression),Unsupervised learning (clustering, dimensionality reduction)

Unit VI : Data Visualization: Principles of effective data visualization, Visualization techniques (scatter plots, histograms, box plots, etc.),Tools and libraries for data visualization, Real-World Applications and Case Studies **06 Hours**

Textbooks

1. "Data Science for Beginners" by Andrew Lee
2. "Introduction to Data Science: A Python Approach to Concepts, Techniques, and Applications" by Laura Igual and Santi Seguí
3. "Data Science for Dummies" by Lillian Pierson

Reference Books

1. "Python for Data Science For Dummies" by John Paul Mueller and Luca Massaron
2. "Data Science from Scratch: First Principles with Python" by Joel Grus
3. "An Introduction to Statistical Learning: with Applications in R" by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani
4. "Practical Statistics for Data Scientists: 50 Essential Concepts" by Peter Bruce and Andrew Bruce

List of Assignments

1. Explore the applications of data science in various industries such as healthcare, finance, retail, etc.
2. Provide examples of how to perform basic operations and manipulations in R.
3. Explain the process of exploratory data analysis (EDA) and its role in the data science process.
4. Explain various data preprocessing techniques such as scaling, normalization, and feature engineering.
5. Explain methods for identifying patterns and trends in data, including correlation analysis.
6. Provide real-world applications and case studies demonstrating the importance and impact of data visualization in various domains.

List of Laboratory Exercises

1. Identify and collect different types of data (structured, unstructured, semi-structured) from online sources or datasets..

2. Create a simple R script to import data from a file, perform basic data cleaning tasks, and generate summary statistics.
3. Use R to generate random samples from different probability distributions (e.g., normal, binomial, Poisson) and visualize the distributions using histograms or density plots.

4. Practical assignment: Fit a simple linear regression model to a dataset using R and interpret the results.
5. Clean a dataset by handling missing values and outliers using R.
6. Compare the performance of different preprocessing techniques on a machine learning task (e.g., classification or regression)..
7. Perform data summarization and visualization on a real dataset using R, including creating scatter plots, histograms, and correlation matrices.
8. Apply unsupervised learning techniques such as clustering or dimensionality reduction to explore patterns in a dataset using R.
9. Use R to visualize real-world datasets and extract insights for decision-making in various domains such as business, healthcare, or environmental science.
10 .mini project

Project Based Learning - Provisional List of Projects

1. Collect customer feedback data from various sources (e.g., surveys, social media) and analyze it to identify trends and patterns.

2. Use R programming to analyze financial data such as stock prices or sales revenue. Apply R basics to import, clean, and visualize the data, and perform basic statistical analysis.

3. Build a predictive model to forecast future trends based on historical data. Use statistical modeling and probability distributions to fit the model and evaluate its performance.

4. Preprocess healthcare data to prepare it for analysis. Apply data cleaning techniques to handle missing values and outliers, and perform feature engineering to extract relevant information.

5. Apply machine learning algorithms for unsupervised learning (clustering) to group customers based on their characteristics.

6. Visualizing Environmental Data

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Value Added Course-I : Introduction to Intellectual Property Rights

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	2 hours/Week	University Examination	Marks	Lecture	2
Practical	hours/Week	Internal Assessment	100 Marks		
		Term Work	-		
		Oral	-		
		Total	100 Marks	Total	02

Course Objective:

The main objective of this course is to make students aware about the types of IP and its protection Process.

Prerequisite:

--

Course Outcomes: On completion of the course, students will have the ability to:

1. Describe the fundamental principles of Intellectual Property and its Rights.
2. Demonstrate the knowledge about Patents and its Filing Process.
3. Demonstrate the need of Copyrights and its Filing Process
4. Describe the Trademarks and its types.
5. Identify the convention for Design Registration
6. Analyse the recent developments in the field of IP.

Unit I Introduction to IP : Meaning of property, Origin, Nature, Meaning of Intellectual Property Rights – Introduction to TRIPS and PCT. Types of Intellectual property rights—Copy Right, Patent, Trade Mark, Trade Secret and trade dress, Design, Layout Design, Geographical Indication, Plant Varieties, Geographical Indication and Traditional Knowledge. **06 Hours**

Unit II Patents Origin, Meaning of Patent, Types, Inventions which are not patentable, Registration Procedure, Rights and Duties of Patentee, Assignment and licence , Restoration of lapsed Patents, Surrender and Revocation of Patents, Infringement, Remedies & Penalties **06 Hours**

Unit III Copyrights Origin, Definition &Types of Copy Right, Registration procedure, Assignment & licence, Terms of Copy Right, Piracy, Infringement, Remedies, Copy rights with special reference to software, Filing Process **06 Hours**

Unit IV TradeMarks: Origin, Meaning & Nature of TradeMarks, Types, Registration of Trade Marks, Infringement & Remedies, Offences relating to Trade Marks, Passing Off, Penalties **06 Hours**

Unit V Design: Meaning, Definition, Object, Registration of Design, Cancellation of Registration, International convention on design, functions of Design. Semiconductor Integrated circuits and layout Design Act-2000. **06 Hours**

Unit VI Developments in IP: Technological and legal Developments in Intellectual Property. Computer Programs: Brief History of the Protection of **06 Hours**

Computer Programs Computer Related Inventions(CRI) Guidelines. Case studies. Information Technology Act

Textbooks

WIPO Intellectual PrcpenyHandbook. Policy, Law and Use (web resource)

Reference Books

1. Patent Law, Handbook Fundamentals of IP for Engineers: K.Bansal & P.Bansal
2. The Patents Act, 1970
3. The Copyright Act 1957
4. The Trademarks Act 1999
5. The Semiconductor Integrated Circuits Layout-design Act, 2000
6. The Information Technology Act, 2000

B.Tech
(Computer Engineering)
Semester- IV

Software Engineering

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Practical:	-	Internal Assessment	40 Marks	Tutorial:	01
Tutorial:	01 Hours/Week	Term Work	-	Practical	00
		Practical	-		
		Total	100 Marks	Total	04

Course Objective:

1. To teach fundamentals of object-oriented concepts and programming
2. To apply the concepts of object-oriented paradigm
3. To design and implement applications for real life problems by using object oriented programming

Prerequisite:

-

Course Outcomes: On completion of the course, students will have the ability to:

1. Demonstrate the Software Engineering approach to software design and development.
2. Apply the Essential processes of SDLC.
3. Demonstrate the knowledge of requirement elicitation by classifying and documenting the requirements
4. Demonstrate software design by modelling artifacts for gathered requirements & analysis.
5. Apply testing strategies and create test cases and test suites
6. Use the project management concepts and tools for managing software project.

Unit I Overview of Software Engineering & Its Methodologies **06 Hours**

Defining Software Engineering, Software Engineering Principles, Software Engineering Ethics, Software Process, Project, Product and People.
Overview of Software development lifecycle methodologies: Waterfall, Agile, Lean, Iterative, Spiral, DevOps.

Unit II Requirements and Design **06 Hours**

Expressing Requirements, Types of Requirements, Feasibility Study, Elicitation Techniques. Requirements Analysis - Structured Analysis, Object Oriented Modelling, Other Approaches. Requirement Specification, Requirement Validation, Requirement Engineering Tools (CASE).
Software Design: Principles of Software Design, Data Design, Architectural Design, Component Level Design, Object-oriented design, Design Notations, User Interface Design

Unit III Coding and Testing **06 Hours**

Coding Guidelines, Coding Methodology, Programming Practice - Top-down, bottom-up, structured programming, information hiding programming, Code verification Techniques, Introduction to No-Code Development approach and tools.

Testing: Software test Characteristics, Test plan, Test Case Design, Testing Strategies, Testing Techniques, Debugging Process, and strategies.

Unit IV Software Quality and Maintenance **06 Hours**

Quality Concepts, ISO 9126 Quality Factors, Mc Call's Quality Factors, SQA plan, SQA Activities, Software Reviews, Sig Sigma & ISO 9000 Quality Standards, capability maturity model, Software Reliability.

Maintenance: Factors affecting software Maintenance, Types of software Maintenance, Software Maintenance Lifecycle.

Unit V SCM and Re-engineering **06 Hours**

Software Configuration Management - Basics, SCM Planning, Project Library, SCM Process - Configuration Identification, Change Control, Version Control, SCM Tools (CASE).

Software re-engineering: Objectives, Principles of Re-engineering, Levels of Abstraction, Software Re-engineering process Model, Business Process Re-Engineering.

Unit VI Software Planning and Cost Estimation **06 Hours**

Responsibility of Software Project Manager, Project Planning, Project Scheduling, People capability maturity model, Risk Management.

Cost Estimation - Basics, Estimation of Resources, Product Cost Factors, Cost Estimation Process, Constructive Cost Model, Function Point Analysis, Decomposition techniques- Problem based Estimation, Process based estimation, use case-based estimation.

Textbooks

- Fundamentals Of Software Engineering, Rajib Mall Phi Learning, 02-Apr-2014, Isbn 8120348982, 9788120348981
- "Software Engineering: Principles and Practices, 2nd Edition by Rohit Khurana, Khurana Rohit · 2010, Vikas Publishing House Pvt Limited", ISBN: 9788125939467
- Software Engineering principles and practices, Rajesh Narang, 2015 McGraw Hill Education, ASINB014ULF4R8

Reference Books

- Software Engineering: A Practitioner's Approach, By Roger Pressman and Bruce Maxim, McGraw Hill, 9th Edition, ISBN10: 1259872971
- Software Engineering, by Ian Sommerville, Pearson; 10th edition, ISBN-10: 0133943038

Project Based Learning - Provisional List of Projects

- Flight Vehicle and Aircraft Systems Engineering.
- Skyscraper
- Software piracy protection system
- E-Learning platform
- Bug tracker
- Railway tracking and arrival time prediction system.

- Employee management system
- Camera motion sensor system
- Operating System task monitoring application
- Data leakage detection system

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Operating System

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Practical:	--	Internal Assessment	40 Marks		
		Term Work	--	Practical	--
		Oral	--		
		Total	100 Marks	Total	03

Course Objective:

The course focuses on the concepts of operating systems enabling students to understand and apply the principles, structure and functioning of Operating system.

Prerequisite:

Data structures and algorithms, Programming Skills.

Course Outcomes: On completion of the course, students will have the ability to:

1. Identify the environment of operating system
2. Apply the concept of process, thread and Inter process communication.
3. Explain the concept of concurrency and deadlocks.
4. Analyze the Storage Management and File System
5. Utilize the concepts of I/O System for communication.
6. Illustrate the use of Network and Distributed System

Unit I Operating System Structure: Introduction, Computer-System Organization, Computer-System Architecture, Operating-System Operations, Operating-System Service, System Calls, Linkers and Loaders, Operating-System Design and Implementation, Operating-System Debugging **06 Hours**

Unit II Process Management: Process Concept, Process Scheduling, Operations on Processes, Interposes Communication, Communication in Client – Server Systems, Threads & Concurrency, CPU Scheduling, PROCESS SYNCHRONIZATION **06 Hours**

Unit III Memory Management: Main Memory, Contiguous Memory Allocation, Paging, Structure of the Page Table, Swapping, Architectures, Virtual Memory **06 Hours**

Unit IV Storage Management and File Systems: Mass-Storage Structure, I/O Systems, File-System Interface, File-System Implementation, File-System Internals, **06 Hours**

Unit V Security and Protection: The Security Problem, Program Threats, System and Network Threats, Cryptography as a Security Tool, User Authentication, Implementing Security Defenses, Goals and Principles of Protection **06 Hours**

Unit VI Networks and Distributed Systems: Network Structure, Distributed File System **Case Studies:** BSD UNIX, MAC System, The Linux System, Windows, RTOS, Android OS **06 Hours**

Textbooks

Operating System Concepts, 10th edition Peter B. Galvin, Greg Gagne, Abraham Silberschatz, John Wiley & Sons, Inc

Reference Books

- Modern Operating Systems -By Andrew S. Tanenbaum (PHI)
- Operating Systems 5th Edition, William Stallings, Pearson Education India
- Design of UNIX Operating Systems, Maurice J. Bach

List of Assignments

1. Write in detail about the Quality (Computer Architecture) based on Features and Functionality of latest OS.
2. It is necessary for an operating system to make efficient use of the computing hardware. When is it appropriate for the operating system to forsake this principle and to "waste" resources? Why is such a system not wasteful?
3. How does the distinction between kernel mode and user mode function as a rudimentary form of protection (security)?
4. Discuss in detail the Concurrency mechanism and Multithreaded programming achieved in latest any OS.
5. Explain the mechanism of process and processor management in Unix/Linux OS
6. Consider a system that supports 5,000 users. Suppose that you want to allow 4,990 of these users to be able to access one file. a. How would you specify this protection scheme in UNIX? Suggest another protection scheme that can be used more effectively for this purpose than the scheme provided by UNIX?

List of Laboratory Exercises

1. Basic functionalities and functions of operating system.
2. Write Shell Script to copy the file system from two directories to a new directory in such a way that only the latest file is copied in case there are common files in both the directories.
3. Implementation of FCFS (First Come First Serve) CPU Scheduling.
4. Implementation of SJF (Shortest Job First) CPU Scheduling.
5. Implementation of Round Robin (RR) CPU Scheduling.
6. Implement Page Replacement Algorithms FIFO and LRU
7. Bankers Algorithm for Deadlock Avoidance
8. Producer Consumer Problem Using Semaphores
9. Implement Virtualization strategy related to resources.

Project Based Learning - Provisional List of Projects

Explore the architecture, features, and functions of open-source operating systems.
2. Design the Processes and thread management with deadlock's, synchronization
3. Design Pre-emptive Priority Scheduling algorithm implementation in any language.

4. Java program to analyze page fault for a given page frame using NRU with paging.
5. The project on simulating the multiprogramming of a specific operating system and dealing with CPU scheduling and Job scheduling.
6. Design the project that computes FCFS, SSTF, and SCAN disk-scheduling algorithms
7. Operating Systems mini project to explore the different algorithms of main memory page replacement
8. Develop a client server application to show the inter process communication.
9. Develop a File System.
10. Write a shell interpreter for LINUX.

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Database Management Systems

TEACHING SCHEME		EXAMINATION SCHEME		CREDIT SCHEME	
					Credits
Lecture:	03 Hours/Week	**End Semester Examination:**60Marks		**Theory**	**03**
Practical:	02 Hours/Week	**Continuous Assessment:40** Marks			
		Term Work:	**25** Marks	**Term Work**	
		Practical:	**25** Marks	**Practical**	**01**
		Total:	**150 Marks**	**Total**	**04**

Course Objective:

- To understand the fundamental concepts of Database Management Systems.
- To acquire knowledge of database query languages and transaction processing.
- To understand systematic database design approaches.
- To acquire the skills to use a powerful, flexible, and scalable general-purpose database to handle Big Data.
- To be familiar with advances in databases and applications

Prerequisite: Discrete Mathematics, Data Structures and Algorithms.

Course Outcomes: On completion of the course, students will have the ability to:

1. Analyze and design Database Management System using ER model.
2. Formulate database queries using database languages.
3. Apply normalization to an ER diagrams.
4. Apply Transaction Management concepts in real-time situations.
5. Use NoSQL databases for processing unstructured data.
6. Differentiate between Complex Data Types and analyze the use of appropriate data types

Unit I - Introduction to Database Management Systems **06 Hours**

Introduction, Purpose of Database Systems, Database-System Applications, View of Data, Database Languages, Database System Structure, Data Models. Database Design and ER Model: Entity, Attributes, Relationships, Constraints, Keys, Design Process, Entity Relationship Model, ER Diagram, Design Issues, Extended E-R Features, converting ER and EER diagram into tables.

Unit II - SQL and PL/SQL **06 Hours**

SQL: Characteristics and Advantages, SQL Data Types and Literals, DDL, DML, DCL, TCL, SQL Operators. Tables: Creating, Modifying, Deleting, Updating. SQL DML Queries: SELECT Query and clauses, Index and Sequence in SQL. Views: Creating, Dropping, Updating using Indexes, Set Operations, Predicates and Joins, Set membership, Tuple Variables, Set comparison, Ordering of Tuples, Aggregate Functions, SQL Functions, Nested Queries.PL/SQL:

Concept of Stored Procedures and Functions, Cursors, Triggers, Assertions, Roles and Privileges.

Unit III -Relational Database Design **06 Hours**

Relational Model: Basic concepts, Attributes and Domains, CODD's Rules. Relational Integrity: Domain, Referential Integrities, Enterprise Constraints. Database Design: Features of Good Relational Designs, Normalization, Atomic Domains and First Normal Form, Decomposition using Functional Dependencies, Algorithms for Decomposition, 2NF, 3NF, BCNF

Unit IV -Database Transaction Management **06 Hours**

Introduction to Database Transaction, Transaction states, ACID properties, Concept of Schedule, Serial Schedule. Serializability: Conflict and View, Cascaded Aborts, Recoverable and Non-recoverable Schedules. Concurrency Control: Lock-based, Time-stamp based Deadlock handling. Recovery methods: Shadow-Paging and Log-Based Recovery, Checkpoints. Log-Based Recovery: Deferred Database Modifications and Immediate Database Modifications.

Unit V - NoSQL Databases **06 Hours**

Introduction to Distributed Database System, Advantages, Disadvantages, CAP Theorem. Types of Data: Structured, Unstructured Data and Semi-Structured Data. NoSQL Database: Introduction, Need, Features. Types of NoSQL Databases: Key-value store, document store, graph, wide column stores, BASE Properties, Data Consistency model, ACID Vs BASE, Comparative study of RDBMS and NoSQL. MongoDB (with syntax and usage): CRUD Operations, Indexing, Aggregation, MapReduce, Replication, Sharding.

Unit VI -Advances in Databases **06 Hours**

Emerging Databases: Active and Deductive Databases, Main Memory Databases, Semantic Databases. **Complex Data Types**: Semi-Structured Data, Features of Semi-Structured Data Models. Nested Data Types: JSON, XML. **Object Orientation**: Object-Relational Database System, Table Inheritance, Object Relational Mapping. **Spatial Data**: Geographic Data, Geometric Data.

Textbooks:

- Database System Concepts by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, 6th Edition, McGraw-Hill Education, 2010.
- "Database Systems" by Connally T, Begg C., Pearson Education, ISBN 81-7808-861-4
- “NoSQL Distilled” by Pramod J. Sadalage and Martin Fowler, Addison Wesley, ISBN10: 0321826620, ISBN-13: 978-0321826626

Reference Books

- C.J. Date, An Introduction to Database Systems, 8/e, Pearson Education, 2004.
- S.K.Singh, "Database Systems: Concepts, Design and Application", Pearson Education, ISBN 978-81-317-6092-5
- Kristina Chodorow, Michael Dierolf, "MongoDB: The Definitive Guide", O'Reilly Publications, ISBN: 978-1-449-34468-9
- Adam Fowler, "NoSQL For Dummies", John Wiley & Sons, ISBN-1118905628
- Kevin Roebuck, "Storing and Managing Big Data - NoSQL, HADOOP and More", Emereopty Limited, ISBN: 1743045743, 9781743045749
- Joy A. Kreibich, "Using SQLite", O'REILLY, ISBN: 13:978-93-5110-934-1
- Ivan Bayross, "SQL, PL/SQL the Programming Language of Oracle", BPB Publications ISBN: 9788176569644, 9788176569644
- Seema Acharya, "Demystifying NoSQL", Wiley Publications, ISBN: 9788126579969

List of Laboratory Exercises:

1) Design a Database and create required tables. For e.g. Bank, College Database
2) Apply the constraints like Primary Key , Foreign key, NOT NULL to the tables.
3) Write a sql statement for implementing ALTER,UPDATE and DELETE.
4) Write the queries to implement the joins
5) Write the query for implementing the following functions: MAX(),MIN(),AVG(),COUNT() ,SUM()
6) Write the query to implement the concept of Integrity constrains
7) Write the query to create the views
8) Perform the queries for triggers
9) Perform the following operation for demonstrating the insertion , updation and deletion using the referential integrity constraints .
10) Write the query for creating the users and their role

Project Based Learning:

1. Medical Health record management system
2. Patient detail management system.
3. Student Management System .
4. On-Demand Online Video Streaming
5. Library Management System.
6. Finances Management System
7. Grocery Management System
8. Weather Management System
9. Web Database system
10. E-commerce Database system.

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Computer Network

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Practical:	02 Hours/Week	Internal Assessment	40 Marks		
		Term Work	25 Marks	Practical	01
		Oral	25 Marks		
		Total	150 Marks	Total	04

Course Objective:

The computer network subject in computer engineering is to equip students with the knowledge and skills necessary to design, implement, and manage computer networks This includes the different types of networks, network devices, protocols, and communication technologies.

Prerequisite:

Data communication concepts

Course Outcomes: On completion of the course, students will have the ability to:

1. Describe the essentials of network models.
2. Compare various Data Link Layer protocols
3. Demonstrate use of various routing algorithms.
4. Distinguish between TCP/IP/UDP.
5. Demonstrate use of various application layer protocols.
6. Discuss world wide web and it's applications.

Unit I: Introduction to Computer Network **06 Hours**

Introduction to Computer Network: Uses of Computer Networks: Business Applications, Home Applications, Mobile Users, Social Issues, Network Hardware: LANs, MANs, WANs. Reference Models: The OSI Reference Model, The TCP/IP Reference Model, Example Networks: The Internet, Connection-Oriented Networks (X.25, Frame Relay & ATM), Ethernet.

THE PHYSICAL LAYER: Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless transmission.

Unit II: THE DATA LINK LAYER **06 Hours**

THE DATA LINK LAYER: Design issues, error detection and correction, elementary data link protocols, sliding window protocols, example data link protocols - HDLC, the data link layer in the internet. THE MEDIUM ACCESS SUBLAYER: Channel allocations problem, multiple access protocols, Ethernet, Data Link Layer switching, Wireless LAN, Broadband Wireless, Bluetooth

Unit III: THE NETWORK LAYER **06 Hours**

Network Layer: Network Layer Design Issues: Store-and-Forward Packet Switching, Services Provided to the Transport Layer, Implementation of Connectionless Service, Implementation of Connection-Oriented Service, Comparison of Virtual-Circuit & Datagram Subnets.

Routing Algorithms: The Optimality Principle, Shortest Path, Routing, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing, Broadcast Routing,

Multicast Routing, Routing for Mobile Hosts.The network layer in the internet (IPv4 and IPv6), Quality of Service.

Unit IV: THE TRANSPORT LAYER **06 Hours**

THE TRANSPORT LAYER: Transport Layer: The Transport Service: Services Provided to the Upper Layers, Transport Service Primitives, Berkeley Sockets. Elements of Transport Protocols: Addressing, Connection Establishment, Connection Release, Flow Control and Buffering, Multiplexing, Crash Recovery, Simple transport Protocol.

User Datagram Protocol (UDP): Introduction to UDP, Remote Procedure Call, The Real-Time Transport Protocol.

Internet Transport Protocols (TCP): Introduction to TCP, The TCP Service Model, The TCP Protocol, The TCP Segment Header, TCP Connection Establishment, TCP Connection Release, Modeling TCP Connection Management, TCP Transmission Policy, TCP Congestion Control, TCP Timer Management, Wireless TCP & UDP Transactional TCP.

Unit V: Application Layer **06 Hours**

Application Layer: The Domain Name System(DNS): The DNS Name Space, Resource Records, Name Servers. Electronic Mail: Architecture & Services, The User-Agent, Message Formats, Message Transfer, Final Delivery.

Application layer protocols: HTTP ,HTTPS ,FTP, SMTP, TELNET SSH, DHCP, SNMP, POP3, IMAP, RTP

Unit VI: World Wide Web **06 Hours**

World Wide Web: Architectural Overview, Static Web Documents, Dynamic Web Documents, HTTP – HyperText Transfer Protocol, Performance Enhancements. Multimedia: Introduction to Digital Audio, Audio Compression, Streaming Audio, Internet Radio, Voice over IP, Introduction to Video, Video Compression, Video on Demand, The MBone – The Multicast Backbone.

Textbooks

- Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI

Reference Books

- An Engineering Approach to Computer Networks-S. Keshav, 2nd Edition, Pearson Education
- Data Communications and Networking – Behrouz A. Forouzan. Third Edition TMH.

List of Laboratory Exercises

1. Familiarization with networking components and devices LAN adapter, Hub, Switches, Routers etc AND Transmission media and tools: Co-axial cable, UTP cable, Crimping tool, Connectors etc.
2. Configure a network using Distance Vector routing Protocol
3. Installation and introduction of simulation tools packet tracer.
4. Introduction to various interior and exterior routing protocols.
5. Study of various LAN topologies and their creation using network devices, cables and Computer. Study of various LAN topologies and their creation using network devices, and implement LAN topologies with packet tracer.

6. Implement Static routing and Default routes by using simulation tools packet tracer
7. Implement RIP by using simulation tools packet tracer
8. configuring IP addresses and DHCP
9. Configure a network using Link State routing Protocol
10. Configure a network using Distance Vector routing Protocol

Project Based Learning - Provisional List of Projects

1. Network Troubleshooting Simulator
2. Home Network Visualization Tool
3. Mobile Data Usage Tracker
4. Network Security Awareness Game
5. Custom VPN Server Setup
6. Network Traffic Analyzer
7. Remote Desktop Control System
8. Smart Home Network Automation

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Web Programming

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Practical:	02 Hours/Week	Internal Assessment	40 Marks		
		Term Work	25 Marks	Practical	01
		Practical	25 Marks		
		Total	150 Marks	Total	04

Course Objective:

The course aims to make students aware of web programming.

Prerequisite:

- Students should have basic knowledge of internet and its security.

Course Outcomes: On completion of the course, students will have the ability to:

1. Understand basic concepts of internet and web server.
2. Apply the concepts of stylesheet.
3. Design webpages using HTML5
4. Understand and apply concepts of table in HTML5
5. Develop an application using basics of JavaScript.
6. Compile basic concepts of PHP

Unit I: Internet and the World Wide Web **06 Hours**

Introduction to internet and its applications, E-mail, telnet, FTP, e-commerce, video conferencing, e-business. Internet service providers, domain name server, internet address, World Wide Web (WWW): World Wide Web and its evolution, uniform resource locator (URL), browsers – internet explorer, Netscape navigator, opera, Firefox, chrome, Mozilla. search engine, web saver – Apache, IIS, proxy server, HTTP protocol

Unit II: HTML 5 **06 Hours**

Introduction, Why HTML5? Formatting text by using tags, using lists and backgrounds, Creating hyperlinks and anchors. HTML Style sheets, CSS formatting text using style sheets, formatting paragraphs using style sheets.

Unit III: HTML 5 Page layout and navigation **06 Hours**

Creating navigational aids: planning site organization, creating text-based navigation bar, creating graphics based navigation bar, creating graphical navigation bar, creating image map, redirecting to another URL, creating

division based layouts: HTML5 semantic tags, creating divisions, creating HTML5 semantic layout, positioning and formatting divisions.

Unit IV: Creating tables **06 Hours**

creating simple table, specifying the size of the table, specifying the width of the column, merging table cells, using tables for page layout, formatting tables: applying table borders, applying background and foreground fills, changing cell padding, spacing and alignment, creating user forms: creating basic form, using check boxes and option buttons, creating lists, additional input types in HTML5, Incorporating sound and video: audio and video in HTML5, HTML multimedia basics, embedding video clips, incorporating audio on web page.

Python file handling: File handling modes, Text & Binary Files – Reading, Writing and Delete.

Error and exception handling: Exceptions, Handling Exceptions, Raising Exceptions, Exception Chaining, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions.

Unit V: Java Script **06 Hours**

Introduction, Client-Side JavaScript, Server-Side JavaScript, JavaScript Objects, JavaScript Security, Operators, delete, new, this, void Statements: Break, comment, continue, delete, do...while, export, for, for...in, function, if...else, import, labelled, return, switch, var, while, with, II Core JavaScript (Properties and Methods of Each) : Array, Boolean, Date, Function, Math, Number, Object, String, regExp Document and its associated objects: document, Link, Area, Anchor, Image, Applet, Layer Events and Event Handlers : General Information about Events, Defining Event Handlers, event.

Unit VI: PHP **06 Hours**

Why PHP and MySQL? Server-side scripting, PHP syntax and variables, comments, types, control structures, branching, looping, termination, functions, passing information with PHP, GET, POST, formatting form variables, super global arrays, strings and string functions, regular expressions, arrays, number handling, basic PHP errors/problems, Advanced PHP and MySQL : PHP/MySQL Functions, Integrating web forms and databases, Displaying queries in tables, Building Forms from queries, String and Regular Expressions, Sessions, Cookies and HTTP, EMail

Textbooks

Web Design The Complete Reference Thomas Powell Tata McGraw Hill

HTML5 Step by Step Faithe Wempen Microsoft Press

PHP 5.1 for Beginners Ivan Bayross Sharanam Shah

Reference Books

I. PHP 6 and MySQL Bible Steve Suehring, Wiley
II. Head First HTML 5 programming Eric Freeman O'Reilly
III. JavaScript 2.0 : The Complete Reference Thomas Powell and Fritz Schneider Tata McGraw Hill

List of Laboratory Exercises

1. Study of search engine, web server and HTTP protocol.
2. Write a program to create basic application using HTML5
3. Write a program to create application using CSS.
4. Write a program to create application using HTML5 Page layout and navigation.
5. Design an application using tables in HTML5.
6. Write a program to perform file handling operations.
7. Create an application using basics of JavaScript.
8. Create an application using basics of Applet.
9. Create an application using PHP and MYSQL
10. Design real time web application.

Project Based Learning - Provisional List of Projects

Data Mining
2. Online Election System Project
3. Advanced Mobile Store
4. Banking Bot Project
5. Farming Assistance Web Service
6. Software Piracy Protection Project
7. Online Diagnostic Lab Reporting System
8. Online Loan Application Verification System
9. Online Herbs Shopping Project
10. Online Bakery Shop System
11. Daily Expense Tracker System
12. Course Material Distribution System
13. Credit Card Fraud Detection Project
14. Smart Health Prediction Using Data Mining
15. Online Furniture Shop Project
16. Computer Assembly Website
17. Car Comparison System Project
18. Cricket Club Management Project
19. Online Blood Bank Project
20. Campus Recruitment System

SBC-IV: Basic Python

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	--Hours/Week	University Examination	-- Marks	Lecture	00
Practical:	02 Hours/Week	Internal Assessment	-- Marks		
		Term Work	25 Marks	Practical	01
		Practical	25 Marks		
		Total	50 Marks	Total	01

Course Objective:

The course aims to make students aware of python programming.

Prerequisite:

Course Outcomes: On completion of the course, students will have the ability to:

1. Demonstrate the knowledge of using data structures in python.
2. Demonstrate the characteristics of object-oriented Python
3. Perform basic operations on file.
4. Understand and implement error and exception handling
5. Design basic GUI using Python Tkinter
6. Implement database connectivity using MySql and SQLite.

Unit I: Python Language Basics **06 Hours**

Python Interpreter, Running I Python Shell, Running Jupyter Notebook, Tab Completion, Introspection, The %run Command, Executing Code from the Clipboard, Terminal Keyboard Shortcuts, About Magic Commands, Language Semantics, Scalar types, Control flow. Data Structures and Sequences – Tuple, List, built in sequence functions, dict, set, strings.

Unit II: Functions, Modules, Packages **06 Hours**

Functions – def statement, returning values, parameters, arguments, local variables, global variables and global statement, Doc Strings, Decorators, lambda, iterators and generators, Modules, Packages.

Unit III: Object-Oriented Approach **06 Hours**

Classes - A simple class, defining methods, the constructor, Member variables, calling methods, Adding inheritance, Class variables, Class methods and static

methods, Properties, Interfaces, New style classes, Doc strings for classes, Private members.

Scope & Namespaces, object, instantiations, Inheritance, Multiple inheritance, Constructors, operator overloading.

Unit IV: File Handling and Error **06 Hours**

Python file handling: File handling modes, Text & Binary Files – Reading, Writing and Delete.Error and exception handling: Exceptions, Handling Exceptions, Raising Exceptions, Exception Chaining, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions.

Unit V: Python Tkinter **06 Hours**

Event Driven Programming, GUI frameworks-Tkinter, Windows and windows attribute, Component, Tk widgets-Ttk and Tix widgets, Geometry Management, Events & Binding Functions, simple GUI applications development.PyGtk – Simple message box, text input dialog box, file selection dialog box.Easy GUI - Example

Unit VI: Backend and Database **06 Hours**

Tornado for windows, building python HTTP web server (GET method), Parameter and Python API (resource and query), Building a JSON GET & POST API, the GET and POST API from JavaScript/HTML

SQLite, MySQL -Environment Setup, Database Connection, CRUD operations.

Textbooks

- Python 3 Object-oriented Programming Second Edition, Dusty Phillips, Packt Publishing
- MySQL for Python: Database Access Made Easy,
- Python GUI Programming with Tkinter, Alan D. Moore, O'Reilly Media, Inc.

Reference Books

- Introduction to Computation and Programming Using Python, John V Guttag, Prentice Hall of India
- Python Essential Reference 4th Edition, David Beazley, Pearson Education.

List of Laboratory Exercises

- Study about Anaconda python software.
- Write a program to understand the control structures of python
- Write a program to learn different types of structures (list, dictionary, tuples) in python
- Write a program to learn concept of functions scoping, recursion, and list mutability.
- Write a program to understand working of exception handling and assertions.
- Write a program to perform basic operations on text files.

- Write a program to implement HTTP server using Python
- Write a program to implement basic GUI application with database connectivity using SQLite
- Write a program to learn GUI programming using Tkinter
- Write a program to implement basic GUI application with database connectivity using MySQL

Project Based Learning - Provisional List of Projects

1. Design and development of Mad Libs generator.

2. Design and development of electronic mail system (Read, write, send and delete operations).

3. Design and development of store billing system.

4. Design and development of typing speed check web application.

5. Design and development of windows application for music player.

6. Design and development of windows Quiz Application.

7. Design and development of web application for daily expense tracker.

8. Design and development of student portfolio management & CV generator system.

9. Design and development of windows based to do list or sticky notes.

10. Design and development of assignment plagiarism checker

Minor Course: Artificial Intelligence and Game Development

Semester -IV: Game Theory

Game Theory					
Teaching Scheme		**Examination Scheme**		**Credit Scheme**	
	Hours/Week		**Marks**		**Credits**
Lecture:	**03 Hours/Week**	**University Examination**	**60 Marks**	**Lecture**	**03**
Practical:	**02 Hours/Week**	**Internal Assessment**	**40 Marks**		
Tutorial:	**00 Hours/Week**	**Term Work**	**25 Marks**	**Practical**	**01**
		Oral	**25 Marks**		
		Total	**150 Marks**	**Total**	**04**

Course Objective:

- Modern artificial intelligence techniques within computer game design and development of interest to students, practitioners, and academics.
- Important and developing industry cases for the development of artificial intelligence techniques in computer games

Prerequisite: OOP, probability, and statistics.

Course Outcomes: On completion of the course, students will have the ability to:

1. Analyze the importance of artificial intelligence in games.
2. Identify different steering behavior in movement of characters
3. Choose the partitioning techniques for designing games
4. Examine decision making methods for games.
5. Design the strategies for games.
6. Select the appropriate design for AI games.

Unit I: Introduction to Game AI **06 Hours**

Role of AI in game design, Game AI interface (movement pathfinding, decision making) complexity (artificial stupidity, intelligent mistakes) game AI inputs, outputs and behaviors.

Unit II: Movement Algorithms **06 Hours**

Position matching, orientation matching, velocity and rotation matching, advanced movement delegation and combination, group movement.

Unit III: Partitioning Algorithms: **06 Hours**

Structure of partitioning algorithms, A* and Heuristics, Abstraction schemes (Lifting and grounding, from partitioning to movement (the steering pipeline, obstacle and collision avoidance)

Unit IV: Decision making Algorithms: **06 Hours**

Structure of decision-making algorithms, decision trees, behavior trees, goal-oriented action planning.

Unit V: Strategy Algorithms **06 Hours**

Structure of strategy algorithm, blackboard, utility theory, Game playing (minmax, Alpha beta pruning), Monte-carlo tree search, coordinated actions (Multi-tier AI, Influence Maps).

Unit VI: Procedural content generation: **06 Hours**

Structure of procedural content generators, pseudorandom number generation, content selection (L-systems, Grammers, randomness) content generation (parametric systems, search based methods)

Textbooks

- Game Theory" Michael Maschler, Eilon Solan, Shmuel Zamir
- Multiagent Systems" Y. Shoham and K. Leyton Brown, Cambridge University Press, online copy available

Reference Books

- Game Theory and Mechanism Design" Y. Narahari, World Scientific and IISc Press – Indian edition available

Laboratory Experiments:

1. Find the winner in nim-game.
2. Finding optimal move in Tic-Tac-Toe using Minimax Algorithm in Game Theory.
3. Find the player who will win by choosing a number in range [1, K] with sum total N.
4. Chessboard Pawn-Pawn game.
5. Coin game of two corners (Greedy Approach)
6. Josephus Problem | (Iterative Solution)
7. A Binary String Game
8. Game of Chocolates | Wythoff's Game

9. Largest, odd divisor Game to check which player wins
10. Game of replacing array elements.

Project Based Learning - Provisional List of Projects

1. Rock Paper Scissor
2. Gambit Play
3. Prisoner's Dilemma
4. The Battle of Sexes
5. The k-server problem

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

****

B. Tech (All Programmes) – 2023 Course

Rules and Regulations

(I) Theory

(A) Theory Examination

Theory examination consists of: (i) End semester examination (ESE), and (ii) Internal assessment (IA).

(i) ESE is of 60 marks for theory courses.

(ii) IA is of 40 marks. Out of 40 marks, 20 marks will be for Unit Tests and 20 marks will be for Project Based Learning for a given course. Two Unit Tests, each of 20 marks, will be conducted. Average of marks obtained in these two unit tests will be considered as UT marks. Roll numbers allotted to the students shall be the examination numbers for the conduction of unit tests.

(B) Standard of Passing

(i) There is a separate passing of 40% of 60 marks, i.e. 24 marks, for ESE for a given course.

(ii) There is a separate passing of 40% of 40 marks, i.e. 16, for IA for a given course.

(iii) A student who fails at ESE in a given course has to reappear only at ESE as a backlog student and clear the head of passing. Similarly, a student who fails at IA in a given course has to reappear only at IA as a backlog student and clear the head of passing

(II) Practical

(A) Practical Examination

Practical examination consists of: (i) Term work, and (ii) Practical/Oral examination for a given course based on term work.

(i) Term work (TW): TW marks are as mentioned in the curriculum structure.

(ii) Practical/Oral (PR/OR): PR/OR marks are as mentioned in the curriculum structure.

(B) Conduction of practical/oral examination

(i) A student will be permitted to appear for practical/oral examination only if he/she submits term work of a given course.

(ii) Practical/oral examination shall be conducted in the presence of internal and external examiners appointed by university.

(B) Standard of Passing

(i) A student shall pass both heads TW and PR/OR separately with minimum 40% of total marks of respective head.

(III) MOOC and Social Activity Course

(i) If a student completes one MOOC during a programme, he/ she will earn an additional TWO credits, subjected to submission of the certificate of completion of the respective course. It is mandatory for a student to complete atleast two MOOC to obtain degree in a given discipline. Students shall register to MOOCs which are offered by any one the following agencies:

(a) SWAYAM : www.swayam.gov.in
(b) NPTEL : www.onlinecourse.nptel.ac.in
(c) Course Era : www.coursera.org
(d) edX online learning : www.edx.org
(e) MIT Open Course ware : www.ocw.mit.edu
(f) Udemy : www.udemy.com
(g) Spoken tutorial : www.spoken-tutorial.org

(ii) If a student completes social activity, he/she will earn additional TWO credits, subjected to submission of the certificate of completion of the respective course/ activity from the relevant authorities. It is mandatory for a student to complete atleast one social activities to obtain degree in a given discipline.

(iv) The additional credits for MOOC and Social Activity will be given only after verification of the authentic document by the Head of the Department and a separate mark-sheet will be submitted by the Head of the Department along with the course examiner

(IV) A. T. K. T

(i) A student who is granted term for B. Tech. Semester-I, III, V, VII will be allowed to keep term for his/her B. Tech. Semester-II, IV, VI, VIII examination, respectively even if he/she appears and fails or does not appear at B. Tech. Semester-I,III, V, VII examination respectively.

(ii) A student shall be allowed to keep term for the B. Tech. Semester-III course if he/she has a backlog of any number of Heads of passing at B. Tech. Semester-I & II taken together.

(iii) A student shall be allowed to keep term for the B. Tech. Semester-V of respective course if he/she has no backlog of B. Tech. Semester-I & II and he/she has a backlog of any number of Heads of passing at B. Tech. Semester-III & IV taken together.

(iv) A student shall be allowed to keep term for the B. Tech. Semester- VII of respective course if he/she has no backlog of B. Tech. Semester-I, II, III, IV and he/she has a backlog of any number of Heads of passing at B. Tech. Semester-V & VI taken together.

(V) Grade Point, Grade Letter and Equivalent Marks

The student must obtain a minimum Grade Point of 5.0 (40% marks) in ESE and also in combined ESE + IA. A student who fails in ESE of a course has to reappear only to ESE as a backlog student and clear that head of passing.

Award of the Class for the Degree considering CGPA: A student who has completed the minimum credits specified for the programme shall be declared to be passed in the programme. The CGPA will be computed every year of all the courses of that year. The grade will be awarded according to the CGPA of every year.

Range of CGPA	**Final Grade**	**Performance Descriptor**	**Equivalent range of Marks (%)**
9.50 ≤ CGPA ≤10.00	O	Outstanding	80 ≤ Marks ≤ 100
9.00 ≤ CGPA ≤ 9.49	A+	Excellent	70 < Marks <80
8.00 ≤ CGPA ≤ 8.99	A	Very Good	60 < Marks < 70
7.00 ≤ CGPA ≤ 7.99	B+	Good	55 < Marks < 60
6.00 ≤ CGPA ≤ 6.99	B	Average	50 < Marks < 55
5.00 ≤ CGPA ≤ 5.99	C	Satisfactory	40 ≤ Marks < 50
CGPA below 5.00	F	Fail	Marks Below 40

(VI) Minor Programme

(i) A students shall receive a MINOR degree when he/she acquire additional 20 credits in a given specialization defined by the UG programmes offered at the institute.

(ii) The theory and practical/oral components for a given course are mentioned in curriculum structure. The theory and examination for a given course are mentioned in Section I and II.

(iii) The grade point, grade letter and equivalent marks system for MINOR programme is mentioned in Section V.

(iv) The MINOR DEGREE programme is OPTIONAL. The interested students may opt MINOR programme.

(v) A student shall complete the MINOR program prior to his/her graduation.