SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
INFORMATION TECHNOLOGY DEPARTMENT

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total credit
							End Sem	Class Work	Sessional Work	End Sem	
IT38001	COMPUTER NETWORKS	3	0	2	3	1	70	30	60	40	4

PER-REQUISITES: Basic Knowledge of Programming.

COURSE OBJECTIVES:

This course is intended to

1. Build an understanding of the fundamental concepts of computer networking.
2. Familiarize the student with the basic taxonomy and terminology of the computer networking area.
3. Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

COURSE OUTCOMES: After completion of course, students will be able to:

CO1 Describe computer networks and concepts of layered approach.
CO2 Explain the functionality and protocols of data link layer.
CO3 Describe the functionality and protocols of network layer.
CO4 Describe transport layer protocols.
CO5 Relate application layer protocols in real world.
CO6 Implement socket programming.

THEORY:

COURSE CONTENTS:

Unit1: Introduction to networks, network v/s distributed systems, Network hardware: broadcast, multicast, peer to peer and point to point. Introduction to LAN, MAN, WAN. Network software: protocol hierarchies, network architecture, protocol stack, Reference models: OSI, TCP/IP reference models, maximum data rate of a channel, transmission media, Switching: circuit, message, and packet switching. Virtual circuits v/s datagram subnets, sockets, DNS, HTTP.

Unit2: Data link layer: framing, error detection, error correction codes. Link layer protocols: simplex, simplex stop and wait, simplex protocol for noisy channel, sliding window, one bit sliding window, go back n, selective repeat, example layers: HDLC, SLIP, PPP.

Unit3: Channel allocation: static, dynamic, Multiple access protocols: pure ALOHA slotted ALOHA, persistent and non-persistent CSMA and CSMA/CD. Collision free protocols: bitmap, binary countdown, limited contention protocol, adaptive tree walk protocol. IEEE 802.1, IEEE 802.2, IEEE 802.3, IEEE 802.4, IEEE 802.5, IEEE 802.6 standards, spanning tree, Interconnections: hubs, link layer switches.

Unit4: Network layer design issues, IPv4 protocol overview, IPv4 packet structure, IPv addressing, subnetting, supernetting, introduction to CIDR, ARP/RARP, ICMP. Optimality principle, Routing algorithms: shortest path, flooding, distance vector, link state,

hierarchical, traffic shaping: leaky bucket, token bucket. Internetworking: Concatenated virtual circuits, connectionless inter networking, NAT, Internetwork routing. Fragmentation: transparent and non-transparent fragmentation.

Unit5: Introduction to Transport layer services, Connectionless transport: UDP, UDP segment structure, Connection oriented transport: TCP, TCP segment structure, RTT estimation, flow control, TCP connection management, Email, file transfer protocols, remote access applications, introduction to firewalls.

TEXT BOOKS RECOMMENDED:

1. Andrews S. Tannen baum ,"Computer Networks", 4th Edition ,Pearson Education.
2. Behrouz A. Forouzen , "TCP/IP protocol Suite" 3rd Edition, Tata McGraw-Hill.
3. William Stallings, "Computer Networking With Internet Protocols And Technology", 3rd Edition, Pearson Education.

REFERENCE BOOKS:

1. BehrouseA. Forouzen , "Data Communication and Networking", 4th Edition, Tata McGraw-Hill.
2. James F. Kurose, Keith W. Ross," Computer Networking: A Top Down Approach" 3rdEditionPearson Education.

ASSESSMENT TOOLS:
Direct assessment: End-Sem Examination, Mid-Term Test, Class Assignments**,** Quiz**,** Attendance
Indirect assessment: Course End Survey

PRACTICAL:
LAB ASSIGNMENTS:
Experiment No.1

1. Describe different types of networks.
2. Describe different types of networking cables
3. Analyze and compare working of networking devices.

Experiment no.2
Use the Virtual Lab of IIT bombay to implement this experiment.

1. Implement cross over and straight through configuration of network.

Use link http://vlabs.iitb.ac.in/vlabs-dev/labs_local/computer-networks/labs/exp1/index.php
Experiment No.3
Use the Virtual Lab of IIT bombay to implement this experiment.

1. Implement peer to peer topology.
2. Implement star topology

Use link http://vlabs.iitb.ac.in/vlabs-dev/labs_local/computer-networks/labs/explist.php

Experiment No. 4
Use the Virtual Lab of IIT bombay to implement this experiment.

1. Implement IPV4 addressing to form a network.
2. Implement subnetting IPV4 subnetting.

Use link http://vlabs.iitb.ac.in/vlabs-dev/labs_local/computer-networks/labs/explist.php

Experiment No.5

Use datagram socket to implement

1. Echo server
2. Date/ Time Server

Experiment No.6

1. Use stream socket to implement
2. Echo Server
3. Date/time server

Experiment No.7

Use Multithreading in Socket Programming to implement chat server

Note: This experiment is to be done in group of 3 student. Design interface for making chat application.

ASSESSMENT TOOLS:

Direct assessment: Lab Assignments, Quiz, Viva-Voce examination (Internal and External), Attendance, Written Test

Indirect assessment: Course End Survey, External Examinar Feedback

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	-	2	-	-
CO2	3	2	3	1	-	-	-	-	-	-	-	-	1	-	-
CO3	3	2	3	1	-	-	-	-	-	-	-	-	1	-	-
CO4	3	3	-	1	-	-	-	-	-	-	-	-	2	-	-
CO5	2	2	-	-	-	2	1	-	-	-	-	-	2	-	-
CO6	3	3	3	3	2	-	-	-	2	1	-	3	2	3	2

SHRI G. S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
INFORMATION TECHNOLOGY DEPARTMENT

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total credit
							End Sem	Class Work	Sessional Work	End Sem	
IT38002	THEORY OF COMPUTATION	3	1	0	4	0	70	30	0	0	4

PRE-REQUISITES: Discrete Structures.

COURSE OBJECTIVES:
This course is intended to

1. Introduce concepts in automata theory and theory of computation.
2. Identify different formal language classes and their relationships.
3. Design grammars and recognizers for different formal languages.
4. Prove or disprove theorems in automata theory using its properties.
5. Determine the decidability and intractability of computational problems.

COURSE OUTCOMES: After completion of the course, students will be able to:

CO1 Differentiate among kind of languages, grammars and automata.

CO2 Design grammars and automata for different language classes and problems.

CO3 Determine the class of the language for a given language or grammar.

CO4 Prove properties of languages, grammars and automata with formal mathematical methods.

CO5 Construct equivalent alternative representations for languages, grammars and automata.

CO6 Describe computability, non-computability, Decidability, undesirability and classes of problems.

THEORY:

COURSE CONTENTS:

UNIT 1: Introduction to theory of computation, Three basic concepts: languages, grammars, and Automata, Finite Automata: DFA and transition graphs, NFA with and without e-moves, equivalence of DFA, NFA without e-moves and NFA with e-moves, reduction of number of states in DFA.

UNIT 2: DFA with output: Mealy and Moore machine, equivalence of Mealy and Moore machine. Regular languages and regular grammars: regular expressions, connection between regular expressions and regular languages, Arden's theorem, regular grammars, Properties of regular languages: closure properties of regular languages, Kleene's theorem, and Pumping Lemma for regular languages.

UNIT 3: Pushdown Automata: context-free grammars, parsing and ambiguity, context-free grammars and programming languages, Simplification of CFG, normal forms of CFGs, membership

algorithm, PDA, DPDA, Equivalence of PDA and CFG, Pumping Lemma for CFLs, closure properties of CFLs.

UNIT 4: Turing machines: Turing's thesis, Turing machines as language accepters, Turing machines as transducers, universal Turing machines, context sensitive languages, linear bounded automata, phrase structured grammar, Chomsky hierarchy, Formal Languages and Automata.

UNIT 5: Recursively enumerable languages, Recursive languages, Partial Recursive functions, Total Recursive Functions, unrestricted grammars. algorithmic computation: decidability and undecidability, Turing machine halting problem, Turing machine models and complexity, class P and NP, basic NP-complete problems.

TEXT BOOKS RECOMMENDED:

1. Peter Linz, "An Introduction to Formal Languages and Automata", 4th Edition Narosa pub. House.
2. Hopcroft, Motwani & Ullman, "Automata Theory, Languages and Computations", 3rd , Pearson Education.

REFERENCE BOOKS:

1. Cohen, "Introduction to Computer Theory", 2nd Edition, Publisher: John Wiley & Sons.
2. K. L. P. Mishra & L. Chandrasekaran, "Theory of Computer Science (Automata, Languages and Computation)", 3rd Edition Second Edition, PHI.
3. Lewis Papadimitrou, "Theory of Computation", Pearson Education.

ASSESMENT TOOLS:

Direct assessment: End-sem examination, mid-semester test, class assignments, attendance
Indirect assessment: Course End Survey

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	1	1										2	1	1
CO2	3	3	3										3	2	1
CO3	3	2											1	1	1
CO4	3	2											1	1	3
CO5	3	2		1									2	2	1
CO6	3	2											1	2	3

SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
INFORMATION TECHNOLOGY DEPARTMENT

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total credit
							End Sem	Class Work	Sessional Work	End Sem	
IT38003	OPERATING SYSTEM	3	-	2	3	1	70	30	40	60	4

PRE-REQUISITES: Basic knowledge of Computer Organization and Architecture.

COURSE OBJECTIVES:
This course is intended to teach:
1. Importance of Operating System.
2. Different types of Operating System and their services, functions and goals.
3. Different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.
4. Virtual memory concepts and disk space allocation.
5. Secondary memory management in operating system.

COURSE OUTCOMES: After completion, of course, students will be able to:
CO1 Define the role of modern operating system in various IT related fields.
CO2 Explain the various modules of modern operating system and their usage.
CO3 Write the various efficient systems related program related to modules of operating system.
CO4 Differentiate various design techniques to implement operating system modules.
CO5 Calculate the performance of various algorithms of operating system modules.
CO6 Discuss the various implementation techniques of operating system modules.

THEORY:
COURSE CONTENTS:

UNIT 1: Introduction to Operating Systems: Functions, Different Types of Operating System. Desirable Characteristics and Features of an Operating System. Operating Systems Services: Types of Services, Different Ways of Providing these Services: Utility Programs, System Calls. Introduction to Processes and Threads, Process Concept, Process State Diagram, Threads, Threads v/s Processes, Advantages of Threads, Implementation of Threads: ULT and KLT , System Calls for Process Management.

UNIT 2: Processes Management: Scheduling Concepts, Types of Schedulers,Scheduling Algorithms, Algorithm Evaluation. Multiple Processor Scheduling. Process Synchronization: Concurrent Processes, Mutual Exclusion, Synchronization, Inter Process Communication, Critical Sections, Locks, Synchronization Hardware, and Semaphores. Classic Problems of Synchronization, Monitors. Deadlocks: Problem, Characterization, Prevention, Avoidance, Recovery. Case Studies Windows, Linux, Solaris etc.

UNIT 3: Memory Management: Different Memory Management Techniques: Partitioning, Swapping, Segmentation, Paging, Segmented Paging and Paged Segmentation, Comparison of These Techniques. Techniques for Supporting the Execution of Large Programs:

Overlays, Dynamic Linking and Loading. Virtual Memory: Concept, Demand Paging, Page Replacement, Thrashing. Case Studies of Linux, Solaris, Windows etc.

UNIT 4: File Systems: File Concept, User's and System Programmer's View of File System. Access Method, Directory Structures, Disk Organization, Different Modules of a File System. Disk-Space Allocation Methods: Contiguous, Linked, and Indexed. Free Space Management, Directory Structures. File Protection. System Calls for File Management. Disk Scheduling Algorithms. Case Studies Ms dos, Unix, Windows, Linux etc.

UNIT 5: Input/Output: Principles and Programming I/O, Input/Output Problem, Asynchronous Operation, Speed Gap, Format Conversion, I/O Interfaces. Program Controlled I/O, Interrupt Driven I/O, Concurrent I/O. Protection and Security: Principal of Protection, Domain of Protection, Access Matrix, Access Control, Capability List. Security Problem, Program Threats, User Authentication.

TEXT BOOKS RECOMMENDED:

1. Abraham Silberschatz, Peter Galvin, and Greg Gagne, "Operating System Concepts", 8th edition, John Wiley & Sons.
2. William Stallings, "Operating Systems", 7th edition, Prentice Hall.
3. Andrew Tannenbaum, "Modern Operating Systems", 3rd edition, Prentice Hall.

REFERENCE BOOKS:

1. Gary Nutt, "Operating Systems", 3rd edition, Addison-Wesley.
2. Deitel, "Operating Systems", 2nd edition, Addison-Wesley.

ASSESSMENT TOOLS :
Direct assessment: End-Sem Examination, Mid-Term Test, Class Assignments**,** Quiz**,** Attendance
Indirect assessment: Course End Survey

PRACTICAL:

LAB ASSIGNMENTS:

Implementing Preemptive Scheduling algorithms.

1. Implementing Non-Preemptive Scheduling algorithms.
2. Implementing deadlock algorithm.
3. Working of User Level Thread using Linux Pthreads Library.
4. Calculating Page Faults using Optimal and LRU page Replacement algorithm.
5. Usage of Binary Semaphore.
6. Implementing fork (), exec () and exit () system calls.
7. Program to understand relation of thread with processes.

ASSESSMENT TOOLS:
Direct assessment: Lab Assignments, Quiz, Viva-Voce examination (Internal and External), Attendance, Written Test
Indirect assessment: Course End Survey, External Examinar Feedback

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	2	2	-	-	-	-	-	-	-	-	-	2	-	-	1
CO2	2	1	1	-	-	-	-	-	-	-	-	2	-	-	1
CO3	2	2	2	2	3	-	-	-	-	-	-	3	2	2	2
CO4	2	2	2	-	-	-	-	-	-	-	-	2	-	-	1
CO5	2	2	2	2	2	-	-	-	-	-	-	2	1	1	1
CO6	2	2	2	2	2	-	-	-	-	-	-	2	1	1	1

SHRI G.S. INSTITUTE OF TECHNOLOGY SCIENCE, INDORE
INFORMATION TECHNOLOGY DEPARTMENT

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total credit
							End Sem	Class Work	Sessional Work	End Sem	
IT38007	DESIGN & ANALYSIS OF ALGORITHMS	3	-	2	3	1	70	30	40	60	4

PRE-REQUISITES: Data Structure, Discrete Structure, Programming Course on C/Java/Python

COURSE OBJECTIVES:
The objectives of this course are:
1. To introduce various designing techniques and methods for algorithms.
2. Provide a solid background for performance analysis of Algorithms using asymptotic and empirical approaches.
3. Provide a thorough knowledge of the most common algorithms and data structures.

COURSE OUTCOMES: The student should be able to:
CO1 Analyze running times of algorithms using asymptotic analysis (time and space complexity).
CO2 Describe various algorithm design techniques.
CO3 Implement major algorithms for the problems of different domain.
CO4 Provide solutions to complex problems using the concept of divide & conquer, dynamic Programming, greedy technique, back tracking and branch & bound techniques.
CO5 Optimize the solution with respect to time complexity & memory usage.
CO6 Describe various classes of algorithms and explain the significance of NP completeness.

THEORY:
COURSE CONTENTS:

UNIT 1: Introduction to algorithms: design and analysis issues, types of algorithms, algorithm specification, Performance analysis: time and space complexity, mathematical preliminaries, functions and their growth rates, recurrence relations and series sums. Review of data structures: stack, queues, trees, dictionaries, priority queue, graphs.

UNIT 2: Divide and conquer approach: merge sort, quick sort, selection sort, Other sorting techniques: bubble sort, insertion sort, heap sort, counting sort, radix and bucket sort, searching minimum and maximum elements, Red-Black trees. Strassen's matrix multiplication, analysis of linear and binary search.

UNIT 3: Dynamic programming: elements of dynamic programming, matrix chain multiplication, parsing, 0/1 knapsack problem, The Greedy Method: elements of greedy method, dynamic programming v/s greedy method, fractional knapsack problem, Graph algorithms: topological sorting, minimum spanning trees, Prims algorithm, Kruskals algorithm: union- find data structure. depth and breadth first search. Strongly connected components. Shortest path algorithm, transitive closure of a graph, all pair shortest path.

UNIT 4: Geometric Algorithms: plane sweep algorithm, Convex Hull. Backtracking and branch and bound techniques: state space, state space tree, 8- Queens Problem, sum of subsets, graph coloring, Hamiltonian cycles: branch and bound methods, 0/1 Knapsack problem. Least cost branch and bound: Traveling Salesmen problem.

UNIT 5: Introduction to parallel algorithms: PRAM algorithms, computation model, fundamental techniques and algorithms, selection, merging and sorting. Classes of algorithm: P, NP, NP-completeness, cooks theorem, NP-Hard, introduction to reduction, satisfiability and 3-CNF SAT, Clique problem, Independent set problem, Vertex cover problem. Introduction to Approximation algorithms: Traveling Salesmen Problem.

TEXT BOOKS RECOMMENDED:

1. T.H. Coreman, C.E. Leiserson and R.L. Rivest, "Introduction to Algorithms", Prentice Hall of India.
2. Jon Kleinberg, Eva Tardos, "Algorithm Design", Pearson Education, LPE.

REFERENCE BOOKS:

1. Elliz Horowitz, SartajSahni, "Computer Algorithms", Pearson Education.
2. Saara Base, "Computer Algorithms: Introduction to Design and Analysis", PE, 2/e, 1988.
3. G.Brassard and P. Brately, "Algorithmic: Theory and Practice", Prentice Hall of India. 1988.

ASSESSMENT TOOLS :

Direct assessment: End-Sem Examination, Mid-Term Test, Class Assignments**,** Quiz**,** Attendance
Indirect assessment: Course End Survey

PRACTICAL:

LAB ASSIGNMENTS:

Experiment 1.

Q.1 Eleven students from your class are selected in Intra-college cricket competition. The heights of students are given in a file Height.txt. If coach wish to arrange the students of the cricket team according to the increasing order of their heights. Implement the -

(i) Selection sort algorithm

(ii) Quick sort algorithm

To help the coach to arrange the students according to their heights.

Q.2 A directed graph G(E,V) consists of V number of vertices and directed edges between nodes. The edge weights are all one. For each node of the graph, two quantities are calculated - the indegree and outdegree. The indegree of the node is the total number of edges incident on the node and outdegree represents the total number of edges going out of the node to the adjacent nodes. Write an efficient algorithm to obtain indegree and outdegree of a directed graph.

Input: Number of nodes in the graph, Adjacent nodes in the graph

Output: Indegree, outdegree of each node of the graph

Experiment 2.

Q.1 Given an array consisting of total 2m elements as per the following format:

{ x1, x2, x3, x4, ….., xm, y1, y2, y3, y4, …., ym }.

Write an efficient program to shuffle the array to {x1, y1, x2, y2, x3, y3, ……, xm, ym } without using extra space. (You can use one temp variable at max for swapping operation)

Q.2 Given two sorted arrays of size m and n respectively, you are required to find the element that would be at the k^{th} final position in the sorted array.

Experiment 3.
Q.1 The problem is to find the closest pair of points in a set of points in x-y plane.
(a) Write a O(n^2) time algorithm by calculating distances of every pair of points and comparing the distances to find the minimum.
(b) Write a Divide and Conquer based algorithm to solve the problem in O(nLogn) time.
Q.2 Given a sorted array of n numbers, give a ɵ(n) algorithm that, given another number x, determines whether or not there exist two elements in the input array whose sum is exactly x.

Experiment 4.
Q.1: In a software company "ABC.com", there are 20,000 employees working. Each employee has different salary than other employee. Write an efficient algorithm to find employee with k^{th} smallest salary. The salary records of the employees are available in a file names "salary.txt" in non-sorted order. You are required to ask the value of "K" from the user.
* Note: The algorithm should have O(n) average time complexity and O(n 2) in worst case. Space complexity should also have good balance with time complexity.
*Hint: Use Divide-and-conquer paradigm to solve.
Two programs - readFile.c and testFile.c are given to you to show how to read file in C and use the readFile.c in testFile.c. On the same lines you are required to use radFile.c in your code to read salary.txt file.
Sample input 1: K=29, Output: 6370
Sample input 2: K=386, Output: 24546
Sample input 3: K=10521, Output: 531623
Sample input 4: K=16822, Output: 845887
Q.2 Let a 1 , . . . , a n be a list of integers. Give an O(n) divide-and-conquer algorithm to find the largest possible sum of a subsequence of consecutive items in the list.
Example: 10 -20 3 4 5 -1 -1 12 -3 1
→ 3 + 4 + 5 + -1 + -1 + 12 = 22

Experiment 5.
Q.1: A Desktop version of Ubuntu running on a machine with x86 architecture. The machine is having 15
processes currently running with each having unique process id. When I run command ps -e on the machine, I get following process id's -
307
39
1289
345
346
768
2918
276
10
762
54
192
300
140
128
Now, let for an array consisting of integers arranged in random order. If we pick two integers randomly from the array at index i and j such that i < j and if elementAt[i] > elementAt[j] then

inversion condition and we need to exchange elementAt[i] and elementAt[j]. The total number of such inversions are required to be counted and the problem is known as counting inversion. Write a divide and conquer based algorithm (θ (n lg n)) to find counting inversions for the processes to be arranged in ascending order on process id for above example of Ubuntu system.
Example of Counting inversion:
2 6 1 4 5 3
1 2 3 4 5 6
The sequence 2, 6, 1, 4, 5, 3 has seven inversions (6,1), (2,1), (6,4), (6,5), (6,3), (5,3), (4,3).
Q.2: The Fibonacci numbers are the numbers in the following integer sequence.
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ……..
Write a program to print 'n' numbers of fibonacci series. The output of the program should also print the time taken by your code (in msec). Two variations of the program are required to be written.
(A) Non- recursive
(B) Divide and conquer approach

Experiment 6:
Q.1: Given an array p[] which represents the chain of matrices such that the i th matrix A i is of dimension p[i-1] * p[i]. Write a function MatrixChainOrder() that returns the minimum number of multiplications needed to multiply the chain. Write a program for implementing matrix chain multiplication (MCM) using divide and conquer approach.
E.g. p[ ] = (3,6,8,2) {This means there are three matrices A 1 , A 2 , A 3 with respective order of matrices (3,6), (6,8), (8,2)}
Possible order of chain multiplications
(A 1 ,(A 2 , A 3)), ((A 1 , A 2)(A 3)), Etc….
Program should find the order of MCM that returns minimum number of multiplications.
Q.2: Given three types of items in three different boxes respectively i.e. each box contains items of same types, with the following respective weights and values:
B = (B 1 , B 2 , B 3)
w i = (2, 3, 4)
v i = (4, 5, 7)
and the knapsack with capacity W = 47. Give a dynamic programming based solution to obtain the optimal solution.

ASSESSMENT TOOLS:
Direct assessment: Lab Assignments, Quiz, Viva-Voce examination (Internal and External), Attendance, Written Test
Indirect assessment: Course End Survey, External Examinar Feedback

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	3	3	1	2	-	-	-	-	-	-	-	1	2	2	2
CO2	2	1	2										1	2	1
CO3	2	3	3	1	1	-	-	3	2	-	-	1	2	2	2
CO4	3	3	3	2	-	-	-	1	3	-	-	1	3	3	3
CO5	3	3	2	3	-	-	-	-	-	-	-	-	1	1	2
CO6	2	2	1	-	-	-	-	-	-	-	-	-	1	1	1

SHRI G.S. INSTITUTE OF TECHNOLOGY SCIENCE, INDORE
INFORMATION TECHNOLOGY DEPARTMENT

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total credit
							End Sem	Class Work	Sessional Work	End Sem	
IT38005	ARTIFICIAL INTELLIGENCE	3	-	-	3	-	70	30	-		3

COURSE OBJECTIVES:

This course is intended to

1. Gain a historical perspective of AI and its foundations.
2. Become familiar with basic principles of AI toward problem solving, inference, perception, knowledge representation, and learning.
3. Investigate applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
4. Experience AI development tools such as an 'AI language', expert system shell, and/or data mining tool.
5. Experiment with a machine learning model for simulation and analysis.
6. Explore the current scope, potential, limitations, and implications of intelligent systems.

COURSE OUTCOMES: Upon successful completion of course students will be able to-

CO1 Demonstrate fundamental understanding of the history of artificial intelligence (AI) and its foundations.

CO2 Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.

CO3 Describe how to build simple knowledge-based systems.

CO4 Apply knowledge representation, reasoning, and machine learning techniques to real world problems

CO5 Apply Game Playing techniques in the given problem.

CO6 Demonstrate fundamental understanding of fuzzy sets operations and properties, fuzzy rules and fuzzy reasoning in the given problem.

THEORY

Course Content

Unit 1: Artificial Intelligence (AI): The AI Problems, The Underlying Assumption, What is An AI Techniques, Characteristics of AI applications, Problem Solving, state space Search and Control Strategies, General problem solving, production systems, forward and backward chaining, exhausive searches: depth first search, breadth first search, Depth limited search, uniform cost search.

Unit 2: Heuristic Search Techniques , Generate and Test, Hill climbing, branch and bound technique, best first search & A* algorithm, AND-OR graphs, problem reduction & AO* algorithm, constraint satisfaction problems, Means-ends analysis.

Unit 3: Knowledge Representation: Representations And Mappings, Approaches To Knowledge Representation, issues, First Order Predicate logic, conversion to clause form,

resolution, unification algorithm, forward and backward reasoning, Semantic Nets, Conceptual Dependency, frames and scripts.

Unit 4: Game Playing: Overview, MiniMax, Alpha-Beta Cut-off, Refinements, Iterative deepening, The Blocks World, Components Of A Planning System, Goal Stack Planning, Nonlinear Planning Using Constraint Posting, Hierarchical Planning, Reactive Systems, Other Planning Techniques.

Unit 5: Fuzzy Sets: fuzzy sets operations and properties, MF formulation and parameterization, fuzzy rules and fuzzy reasoning, fuzzy inference systems, fuzzy models. Introduction to genetic algorithm.

Text Books

1. Rich & Knight , "Artificial Intelligence" ,2nd Edition,Tata Mcgraw Hill.
2. Rajsekaran and Pai, "Neural Network and Fuzzy Logic", 2nd Edition,Prentice Hall of India.

Reference Books

1. Russel & Norvig, "Artificial Intelligence",2nd Edition ,Pearson Education.
2. Patterson, "Introduction to AI and expert System",2nd Edition, Prentice Hall of India.
3. S.N. Sivananadan & S.N. Deepa, "Principles of Soft Computing", 1st Edition,Wiley India.

ASSESSMENT TOOLS :
Direct assessment: End-Sem Examination, Mid-Term Test, Class Assignments**,** Quiz**,** Attendance
Indirect assessment: Course End Survey

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	–	–	–	–	–	–	–	–	–	–	–	-	-	-
CO2	3	–	–	–	–	–	–	–	–	–	–	–	2	-	2
CO3	3	–	–	–	–	1	–	–	–	–	–	–	-	-	2
CO4	3	2	3	–	3	3	3	–	–	–	–	3	2	2	3
CO5	3	3	3	2	2	2	–	–	–	–	–	2	1	2	3
CO6	3	3	3	3	–	–	–	–	–	–	–	–	2	2	3
Average	3	2.3	3	2.5	2.5	2.5	3	-	-	-	-	2.5			

SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
DEPARTMENT OF INFORMATION TECHNOLOGY

S. No.	Code No	Subject Name	L	T	P	Th. Credits	Pr. Credits	Marks				
								TH	Class Work	Sessional Work	PR	Total
1	IT38452	System Software Laboratory-III			2		1			40	60	100

PRE-REQUISITES: None.

COURSE OBJECTIVES:
To Gain skills for systems programming including file system programming.

COURSE OUTCOMES: After the completion of the course, the students will be able to:-
CO1 Demonstrate the installation process of operating system and application software.
CO2 Explain different components of system software.
CO3 Describe Linux operating commands and their execution.
CO4 Demonstrate the installation process of different server.
CO5 Describe the Process management commands and their execution.
CO6 Describe network configuration commands.

THEORY:
COURSE CONTENTS:

UNIT 1: System Administration in Linux Operating Systems: Partition management with linux and Installation of Linux, Post Install, Configuring run levels and consoles. Fstab and Crontab, file system, file and directory, Commands permissions, users, groups, devices, Linux services and daemons, searching with locate, updatedb.

UNIT 2: Concept of modules, Linux Backup and Recovery Services, kernel compilation, Compilation options, creating initial Ram disk image, Patch addition and module configuration in Linux. Extra installations using RPMs and Make utility, creating a Local Area Network using Red Hat Linux, Assigning static IP addresses, connecting to Internet.

UNIT 3: Network services, Network Security, Creating Linux Network-Dynamic IP addresses, DHCP server and Client Configuration, Host names, Name lookup on LAN. Filters and scripting for management of server using sed, grep, awk.

UNIT 4: NFS-Native file sharing service, Configuring NFS, Samba server, Creating Samba share and Samba users, Configuring Samba windows client Configuring FTP server and Client. Telnet and SSH, Running Telnet and SSH sessions, Configuring CUPS, NIS, DNS, Apache, SQUID, IPTABLES. Remote management and administration. Backup and Recovery Services

UNIT 5: Database Administration: creation of database and its user, granting rights to users on tables. Database storage methods, backup and recovery. GUI development under LINUX: Qt, GUI Design: Characteristics, Essentials and Implementation, Wine utility configuration.

TEXT BOOKS RECOMMENDED:

1. SumitabhaDas, “Unix: Concepts and Application”, Fourth Edition Tata McGraw Hill.
2. Richard Petersen, “Linux: The Complete Reference”, Sixth Edition Tata McGraw Hill.

REFERENCE BOOKS:

1 Aeleen Frisch, “Essential System Administration”, 3e O’Reilly Publication.
2 Christopher Negus, “Linux Bible”, 9th Edition Wiley Publication.

PRACTICAL:
LAB ASSIGNMENTS:

1. To Install and configure Linux Operating System and Windows 10 Professional OS to desk top computer and Understand control panel settings.
2. To Install and configure Microsoft Advanced Server installation.
3. To Install and configure Samba Server installation.
4. To Install and configure Network Services and DHCP Server.
5. To Install and configure Remote Management(RPC)
6. Perform Database Administration Command.
7. To Install and configure Android Studio installation.
8. Installing and uninstalling an Application Software in LINUX Environment.
9. Perform and execute network configuration commands.
10. Perform and execute process management commands.

ASSESSMENT TOOLS:
Direct assessment: Lab Assignments, Quiz, Viva-Voce examination (Internal and External), Attendance, Written Test
Indirect assessment: Course End Survey, External Examinar Feedback

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or ‘-’):

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	3	2	1	1	1	-	-	-	1	-	-	1	2	2	1
CO2	3	3	1	1	1	-	-	-	-	-	-	1	2	2	1
CO3	3	3	1	1	1	-	-	-	-	-	-	1	2	2	1
CO4	3	2	1	1	1	-	-	-	1	-	-	1	2	2	1
CO5	3	2	1	1	1	-	-	-	-	-	-	1	2	2	1
CO6	3	2	1	-	-	-	-	-	-	-	-	1	2	2	1