Proposed Scheme & Syllabus of UG Engineering Program

Bachelor of Technology (B.Tech.)

Computer Science & Engineering

2020-21

IPS ACADEMY
INSTITUTE OF ENGINEERING & SCIENCE, INDORE

(A UGC Autonomous Institute affiliated to RGPV)

New Scheme of Examination as per AICTE Flexible Curricula

V Semester Bachelor of Technology (B.Tech) Computer Science & Information Technology *(w.e.f. July,2020)*

S.No.	Subject Code	Category	Subject Name	Maximum Marks Allotted					Total Marks	Contact Hours per week			Total Credits
				Theory			Practical						
				End Sem.	Mid Sem. Exam.	Quiz/ Assignment	End Sem	Term work / Lab Work & Sessional		L	T	P	
1.	CSIT-501	DC	Computer Networking	70	20	10	30	20	150	3	-	2	4
2.	CSIT -502	DC	Operating System	70	20	10	30	20	150	3	-	2	4
3.	CSIT -503	DE	Departmental Elective	70	20	10	-	-	100	3	1	-	4
4.	CSIT -504	OE	Open Elective	70	20	10	-	-	100	3	-	-	3
5.	CSIT -505	D Lab	Lab (Shell Programming)	-	-	-	30	20	50	-	-	4	2
6.	CSIT -506	O/E Lab	(Open Source Software Lab)	-	-	-	30	20	50	-	-	4	2
7.	CSIT -507	IN	Evaluation of Internship-II	-	-	-	-	100	100	-	-	6	3
8.		IN	Internship - III	To be completed anytime during Fifth/Sixth semester. Its evaluation/credit to be added in Seventh Semester.									
9.	CSIT -508	P	Minor Project- I	-	-	-	-	50	50			4	2
10.	Additional Credits[#]	*[#]Additional credits can be earned through successful completion of credit based MOOC's Courses available on SWAYAM platform (MHRD) at respective UG level.*											
			Total	280	80	40	120	230	750	12	1	22	24

Departmental Electives	Open Electives
CS 503 (A) Theory of Computation	CS 504 (A) Cyber Security
CS 503 (B) Microprocessor & Interfacing	CS 504 (B) Artificial Intelligence
CS 503 (C) Principles of programming Languages	CS 504 (C) Pattern Recognition

1 Hr Lecture	1 Hr Tutorial	2 Hr Practical
1 Credit	**1 Credit**	**1 Credit**

RAJIV GANDHI PROUDYOGIKI

VISHWAVIDYALAYA, BHOPAL New Scheme

Based On AICTE Flexible Curricula

Computer Science & Information Technology, V-Semester

CSIT-501 Computer Networking

Objectives:

1. To introduce concepts and fundamentals of data communication and computer networks.
2. To explore the inter-working of various layers of OSI.
3. To explore the issues challenges of protocols design while delving into TCP/IP protocol suite.
4. To assess the strengths and weaknesses of various routing algorithms.
5. To understand the transport layer and various application layer protocols.

Course Outcome: After the completion of this course, the students will be able to:

1. Characterize and appreciate computer networks from the view point of components and from the view point of services.
2. Display good understanding of the flow of a protocol in general and a network protocol in particular.
3. Model a problem or situation in terms of layering concept and map it to the TCI/IP stack.
4. Select the most suitable Application Layer protocol (such as HTTP, FTP, SMTP, DNS, Bit torrent) as per the requirements of the network application and work with available tools to demonstrate the working of these protocols.
5. Design a Reliable Data Transfer Protocol and incrementally develop solutions for the requirements of Transport Layer.
6. Describe the essential principles of Network Layers and use IP addressing to create Subnets for any specific requirements.

Unit-I

Introduction to computer networks: Network – Component and Categories – Topologies Reference Models: ISO/OSI Model and TCP/IP Protocol suite. Principals of physical layer: Transmission Media, Bandwidth, Multiplexing, Switching.

UNIT II

Data Link Layer: Need, Services Provided, Framing, Flow Control, Error control. Data Link Layer Protocol: Elementary &Sliding Window protocol: 1-bit, Go-Back-N, Selective Repeat, Hybrid ARQ. Protocol verification: Finite State Machine Models & Petri net models. ARP/RARP.

UNIT III

MAC Sub layer: MAC Addressing, Binary Exponential Back-off (BEB) Algorithm, Distributed Random Access Schemes/Contention Schemes: for Data Services (ALOHA and Slotted ALOHA), for Local-Area Networks (CSMA, CSMA/CD, CSMA/CA), Collision Free Protocols: Basic Bit Map,Binary Count Down,Adaptive Tree Walk, Performance Measuring Metrics. IEEE

Standards 802 series & their variant.

UNIT IV

Network Layer: Need, Services Provided, Design issues, Routing algorithms: Least Cost Routing algorithm, Dijkstra's algorithm, Bellman-ford algorithm, Hierarchical Routing, Broadcast Routing, Multicast Routing. IP Addresses, Header format, Packet forwarding, Fragmentation and reassembly, ICMP, Comparative study of IPv4 & IPv6.

UNIT V

Transport Layer: Design Issues, UDP: Header Format, Per-Segment Checksum, Carrying Unicast/Multicast Real-Time Traffic, TCP: Connection Management, Reliability of Data Transfers, TCP Flow Control, TCP Congestion Control, TCP Header Format, TCP Timer Management.Application Layer: WWW and HTTP, FTP, SSH, Email (SMTP, MIME, IMAP), DNS, Network Management (SNMP).

Recommended Books:

1. Andrew S. Tanenbaum, David J. Wetherall, "Computer Networks" Pearson Education.
2. Douglas E Comer, "Internetworking WithTcp/Ip Principles, Protocols, And Architecture-Volume I" 6th Edition,Pearson Education.
3. DimitriBertsekas, Robert Gallager, "Data Networks", PHI Publication, Second Edition.
4. KavehPahlavan, Prashant Krishnamurthy, "Networking Fundamentals", Wiley Publication.
5. Uyless Black, "Computer Networks", PHI Publication, Second Edition.
6. Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, "Computer Networks: An Open Source Approach", McGraw Hill.

List of Experiments:

1. Study of Different Type of LAN& Network Equipments.
2. Study and Verification of standard Network topologies i.e. Star, Bus, Ring etc.
3. LAN installations and Configurations.
4. Write a program to implement various types of error correcting techniques.
5. Write a program to Implement various types of framing methods.
6. Study of Tool Command Language (TCL).
7. Study and Installation of Standard Network Simulator: N.S-2, N.S3.OpNet,QualNetetc .
8. Study & Installation of ONE (Opportunistic Network Environment) Simulator for High Mobility Networks .
9. Configure 802.11 WLAN.
10. Implement &Simulate various types of routing algorithm.
11. Study & Simulation of MAC Protocols like Aloha, CSMA, CSMA/CD and CSMA/CA using Standard Network Simulators.
12. Study of Application layer protocols-DNS, HTTP, HTTPS, FTP and TelNet.

RAJIV GANDHI PROUDYOGIKI

VISHWAVIDYALAYA, BHOPAL New Scheme

Based On AICTE Flexible Curricula

Computer Science and Information Technology, V - Semester
CSIT 502 OPERATING SYSTEM

Objective:
The purpose of this subject is to cover the underlying concepts of Operating System. This syllabus provides a comprehensive introduction of Operating System, Process Management,Memory Management, File Management and I/O management.

Course Outcomes: After the completion of this course, the students will be able to:

1. Understand the core concepts of operating system, evolution and types of operating system.
2. Understand CPU and Process scheduling, inter-process communication and deadlock.
3. Understand various memory management techniques.
4. Understand the Basic concept of input output organization.
5. Describe the concept of file and device management.

UNIT 1
Introduction to Operating Systems:
Function, Evolution, Different Types, Desirable Characteristics and features of an O/S, Operating Systems Services: Types of Services, Different ways of providing these Services Utility Programs, System Calls, fork ().

UNIT 2
CPU Scheduling:
Process Concept, Scheduling Concepts, Types of Schedulers, Process State Diagram, Scheduling Algorithms, Algorithms Evaluation, System calls for Process Management; Multiple Processor Scheduling, Concept of Threads.
Concurrent Processes: Real and Virtual Concurrency, Mutual Exclusion, Synchronization, Inter- Process Communication, Critical Section Problem, Solution to Critical Section Problem: Semaphores – Binary and Counting Semaphores, WAIT & SIGNAL Operations And their implementation. Deadlocks: Deadlock Problems, Characterization, Prevention, Avoidance, Recovery.

UNIT 3
Memory Management:
Different Memory Management Techniques – Partitioning, Swapping, Segmentation, Paging, Paged Segmentation, Comparison of these techniques, Techniques for supporting the execution of large programs: Overlay, Dynamic Linking and Loading, Virtual Memory – Concept, Implementation by Demand Paging etc.

UNIT 4:

Input / Output:

Principles and Programming, Input/Output Problems, Asynchronous Operations, Speed gap Format conversion, I/O Interfaces, Programme Controlled I/O, Interrupt Driven I/O, Concurrent I/O. Introduction to Network, Distributed and Multiprocessor Operating Systems. Case Studies: Unix/Linux, WINDOWS

UNIT 5:

File Systems:

File Concept, User's and System Programmer's view of File System, Disk Organization, Tape Organization, Different Modules of a File System, Disk Space Allocation Methods – Contiguous, Linked, Indexed. Directory Structures, File Protection, System Calls for File Management, Disk Scheduling Algorithms.

Recommended Books:

1. Silberschatz, Galvin, Gagne, "Operating System Concepts'',
2. Wiley, 9/E 2. William Stalling, "Operating Systems", Pearson Education
3. Andrew S. Tanenbaum, "Modern Operating Systems", 3/e, Prentice Hall
4. Maurice J. Bach, " The Design of Unix Operating System", Prentice Hall of India,
5. Bovet & Cesati, "Understanding the Linux Kernel", O'Reily, 2/E.

List of Experiments:

1. Case Study of different Operating Systems and evaluation of Operating System.
2. Comparative study of different operating system
3. Write a program to implement FCFS CPU scheduling algorithm
4. Write a program to implement SJF CPU scheduling algorithm
5. Write a program to implement Priority CPU Scheduling algorithm.
6. Write a program to implement Round Robin CPU scheduling algorithm
7. Write a program to implement Banker's algorithms.
8. Write a program to implement classical inter process communication problem(producer consumer).
9. Write a program to implement various page replacement algorithms
10. Write a program to implement various Disk scheduling Algorithms.

RAJIV GANDHI PROUDYOGIKI

VISHWAVIDYALAYA, BHOPAL New Scheme

Based On AICTE Flexible Curricula

Computer Science and Information Technology, V - Semester
Departmental Elective CSIT- 503 (A) Theory of Computation

Course Objectives

1. Student learns some fundamental concepts in automata theory and designing of Finite Automata, conversion NFA to DFA. Application of Finite Automata in computer science and real world.
2. Obtain minimized DFA and Application of regular expression and conversion from RE to Finite Automata and Finite Automata to Regular Expression and Proving language are not regular.
3. Designing of CFG's , Construction of parse trees, finding and removing ambiguity in grammars, simplification of CFG, Conversion of grammar to Chomsky Normal Form.
4. Designing problems on Pushdown Automata and conversion of grammar to PDA, PDA to Grammar.
5. Designing Turing machines, understanding the working of various types of Turing machines and studyP and NP type problem.

Course Outcomes: After the completion of this course, the students will be able to:

1. Convert between finite automata, regular grammars, and regular expression representations of regular languages
2. Convert between grammars and push-down automata for context-free languages
3. Translate a context-free grammar from one form to another
4. Understand Push down automata
5. List examples of un-decidable problems

UNIT I

Introduction of the theory of computation, Finite state automata – description of finite automata, properties of transition functions, Transition graph, designing finite automata, FSM, DFA, NFA, 2-way finite automata, equivalence of NFA and DFA, Mealy and Moore machines.

UNIT II

Regular grammars, regular expressions, regular sets, closure properties of regular grammars, Arden's theorem, Myhill-Nerode theorem, pumping lemma for regular languages, Application of pumping lemma, applications of finite automata, minimization of FSA.

UNIT III

Introduction of Context-Free Grammar - derivation trees, ambiguity, simplification of CFGs, normal forms of CFGs- Chomsky Normal Form and Greibach Normal forms, pumping lemma for CFLs, decision algorithms for CFGs, designing CFGs, Closure properties of CFL's.

UNIT IV

Introduction of PDA, formal definition, closure property of PDA, examples of PDA, Deterministic Pushdown Automata, NPDA, conversion PDA to CFG, conversion CFG to PDA.

UNIT V

Turing machines - basics and formal definition, language acceptability by TM, examples of TM, variants of TMs – multitape TM, NDTM, Universal Turing Machine, offline TMs, equivalence of single tape and multitape TMs. Recursive and recursively enumerable languages, decidable and undecidable problems – examples,halting problem, reducibility. Introduction of P, NP, NP complete, NP hard problems and Examples of these problems.

Recommended Books:

1. Daniel I.A. Cohen,"Introduction to Computer Theory",Wiley India.
2. John E Hopcroft, Jeffrey D. Ullman and Rajeev Motwani, "Introduction to Automata Theory, Languages and Computation", Pearson Education.
3. K.L.P Mishra & N.Chandrasekaran,"Theory of Computer Science", PHI Learning.
4. Peter Linz, "Introduction to Automata Theory and Formal Languages", Narosa Publishing.
5. John C Martin, "Introduction to languages and the theory of computation", TATA McGraw Hill.

RAJIV GANDHI PROUDYOGIKI

VISHWAVIDYALAYA, BHOPAL New Scheme

Based On AICTE Flexible Curricula

Departmental Elective CSIT- 503 (B) Microprocessor and Interfacing

Objectives:

1. To introduce basic concepts of microprocessor
2. To introduce serial and parallel bus standards.
3. To introduce programming in assembly language.
4. To introduce basic concepts of interfacing memory and peripheral devices to a microprocessor.

Course Outcomes:

1. Explain the microprocessor's and Micro controller's internal architecture
2. Apply knowledge and demonstrate programming proficiency using the various addressing modes and data transfer instructions of the target microprocessor and microcontroller.
3. Compare accepted standards and guide lines to select appropriate Micro processor (8085&8086) and Microcontroller to meet specified performance requirements.
4. Analyze assembly language programs
5. Design electrical circuitry to the Microprocessor I/O ports in order to interface the processor to external devices.Evaluate assembly language programs

UNIT –I

Evolution of microprocessor, single chip micro computers, Micro processor Application, Microprocessor and its architecture, addressing modes, instruction, Instruction sets, Arithmetic and Logic Instruction, Program control instruction, Introduction –8086 family, procedure and macros, connection , Timing and Troubleshooting interrupt, 80286, 80836 and 80486 micro processor system concept.

UNIT –II

Microprocessor Cycle, AIU, Timing and control Unit, Register data, Address bus, Pin Configuration, Intel 8086 instruction, Opcode and operands, limitation word size. Programming the microprocessor Assembly language, The Pentium and Pentium Pro Micro Processor with features, Pentium II, Pentium III and Pentium –IV Microprocessor with software changes. Instruction set for Intel 8086, Introduction Intimation and data formats, Addressing modes, Status flags, Symbols and abbreviations, programming of microprocessors, Assembly language, high level language, areas of application of various languages, Stacks, Sub routines system, software, commands in assembly language, software Development, Debugging program, Modular programming, Structured programming, Top-down, Bottom-up design , MACRO microprogramming.

UNIT-III

Assembly language programming with Examples like Addition of 8/16-bit Binary number, subtraction of 8/16 bit binary number, Address partitioning, addressing mode, type of addressing mode, memory and I/o interfacing, Data transfer schemes, Interfacing device and I/o devices I/o ports, Basic I/o Interfacing MDS, Micro controllers, I/o processor and co-processors ,Microcomputer Development system, Single chip micro computers, intel 8748 intel 8051, inter 8096, intel 8049intel 2920/2921, I/o processor UPI-425,UPI-41,42, Co- processor, math processor math co-processor –8087, 80287, 80387DX 803875x

UNIT –IV

Bus Interface I/o port Addressing, decoding 8279, Programmable key board/display interface, 8254 Internal Timer, 16550 programmable communication interface A/D, 8259A Programmable Interrupt Controller, 8237 DMA Controller, Shared bus operation, disk Memory system Video display. ISA Bus, Extended ISA (EISA) and VESA Local Buses, Peripheral Component Inter Connect (Pc I) Bus, Parallel Printer interface (LPT) Universal serial Bus (USB) Accelerated graphics port (AGP),Programmable Communication interfere 8251 VSART CRT Controller 8275, 6854, Floppy disk Controller 8272, I/o processor 8089.

UNIT –V

Memory Unit, RAM,SRAM, DRAM,ROM, PROM EPROM, EEPROM Nonvolatile RAM semiconductor Technology for memory, Shift register, Magnetic Memory, Tap, disc, main memory and secondary memory cache memory, program memory and Data Memory, Real and virtual memory Buses, memory Addressing capacity of CPU, processing speed of computer

Recommended Books:

1. Douglas V Hall, "Microprocessors and interfacing –Programming & Hardware" TMH
2. Barry B. Brey, "The intel Microprocessor –8086", Pearson Education
3. Kenneth J.Ayala,"The 8086 Microproccessor: Programming & Interfacing The PC",CengageLearning
4. Krishna Kant,"Microprocessors and Microcontrollers", PHI Learning
5. A.K.Ray KM Bhurchandi, "Advanced Microprocessor and peripherals" McGraw Hill
6. R.S. Gaonkar ,"Microprocessors and interfacing", TMH

List of Experiments:

1. Assembly Language Programs of Microprocessor 8086.
2. Assembly Language Programs of Microcontroller 8051.
3. Assembly Language Programs for Interfacing Chips
4. To study 8085 microprocessor system.
5. Write a program using 8085 and verify for subtraction of two 8 bit hex numbers and store result at 2050H.
6. Write a program using 8085 and verify for addition of two 8 bit numbers and store the result at 2002H.
7. Write a program using 8085 and verify for:

a) Addition of two 16 bit number using register pair.
b) Addition of two 16 bit number directly..

RAJIV GANDHI PROUDYOGIKI

VISHWAVIDYALAYA, BHOPAL New Scheme

Based On AICTE Flexible Curricula

Computer Science and Information Technology, V - Semester
Departmental Elective CSIT- 503 (C) Principles of Programming Languages

Objective:

1. To introduce the major programming paradigms, and the principles and techniques involved in design and implementation of modern programming languages.
2. To introduce notations to describe syntax and semantics of programming languages.
3. To analyze and explain behavior of simple programs using concepts such as binding, scope, control structures, subprograms and parameter passing mechanisms.
4. To introduce the concepts of concurrency control and exception handling

Course Outcomes: After the completion of this course, the students will be able to:

1. Have the background for choosing appropriate programming languages for certain classes of programming problems
2. Be able to program in an imperative (or procedural),an object-oriented, a functional, and a logical programming language
3. Understand the significance of an implementation of a programming language in a compiler or interpreter
4. Have the ability to learn new programming languages
5. Be able to design a new programming language

UNIT-I

Language Evaluation Criteria, influences on Language design, Language categories, Programming Paradigms – Imperative, Object Oriented, functional Programming, Logic Programming. Programming Language Implementation –Compilation and Virtual Machines, programming environments

UNIT-II

Data types: Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types, design and implementation uses related to these types. Names, Variable, concept of binding, type checking, strong typing, type compatibility, named constants, variable initialization, Sequence control with Expressions, Conditional Statements, Loops, Exception handling.

UNIT-III

Subprograms and Blocks: Fundamentals of sub-programs, Scope and lifetime of variable, static and dynamic scope, Design issues of subprograms and operations, local referencing environments, parameter passing methods, overloaded sub-programs, generic sub-programs, design issues for functions overloaded operators, co routines.

UNIT-IV

Abstract Data types: Abstractions and encapsulation, introductions to data abstraction, Static and Stack-Based Storage management. heap based storage management. Garbage Collection. Object oriented programming in small talk, C++, Java, C#, PHP, Perl . Concurrency: Subprogram level concurrency, semaphores, monitors, massage passing, Java threads, C# threads

UNIT–V

Exception handling, Exceptions, exception Propagation, Exception handler in C++ and Java. Logic Programming Language : Introduction and overview of logic programming, basic elements of prolog, applicationof logic programming. Functional Programming Languages: Introduction, fundamentals. Introduction to 4GL.

Recommended Books:

1. Sebesta,"Concept of programming Language", Pearson Edu.
2. Louden, "Programming Languages: Principles & Practices", Cengage Learning
3. Tucker, " Programming Languages: Principles and paradigms", Tata McGraw –Hill Terrance W Pratt, "Programming Languages: Design and Implementation", Pearson Edu.
4. CavloGhezzi& Mehdi Jazayeri" Programming Languages

Concepts", WilleyIndia 5. EHorowitz, "Programming
Languages", 2nd Edition, Addison Wesley

List of Experiments:

1. Define a LISP function to compute sum of squares.
2. Define a LISP function to compute difference of squares.(if x > y return x2 -y2, otherwise y2 -x2).
3. Define a Recursive LISP function to solve Ackermann's Function.
4. Define a Recursive LISP function to compute factorial of a given number.
5. Define a Recursive LISP function which takes one argument as a list and returns last element of the list. (Do not use last predicate).
6. Define a Recursive LISP function which takes one argument as a list and returns a list except last element of the list. (Do not use but last predicate).
7. Define a Recursive LISP function which takes one argument as a list and returns reverse of the list. (Do not use reverse predicate).
8. Define a Recursive LISP function which takes two arguments first, an atom, second, a list, returns a list after.

RAJIV GANDHI PROUDYOGIKI

VISHWAVIDYALAYA, BHOPAL New Scheme

Based On AICTE Flexible Curricula

Computer Science and Information Technology, VI - Semester
Departmental Elective CSIT- 504 (A) Cyber Security

Objective:

1. Exhibit knowledge to secure corrupted systems, protect personal data, and secure computer networks in an Organization.
2. Understand key terms and concepts in Cryptography, Governance and Compliance
3. Understand principles of web security and to guarantee a secure network by monitoring
4. Analyzing the nature of attacks through cyber/computer forensics software/tools.

Course Outcomes: After the completion of this course, the students will be able to:

1. Analyze and evaluate the cyber security needs of an organization.
2. Determine and analyze software vulnerabilities and security solutions to reduce the risk of exploitation.
3. Measure the performance and troubleshoot cyber security systems.
4. Comprehend and execute risk management processes, risk treatment methods, and key risk and performance indicators
5. Able to Design and develop security architecture for an organization, operational and strategic cyber security strategies and policies.

UNIT 1

Introduction of Cyber Crime, Challenges of cyber crime, Classifications of Cybercrimes: E- Mail Spoofing, Spamming, Internet Time Theft, Salami attack/Salami Technique,

UNIT 2

Web jacking, Online Frauds, Software Piracy, Computer Network Intrusions, Password Sniffing, Identity Theft, cyber terrorism, Virtual Crime, Perception of cyber criminals: hackers, insurgents and extremist group etc. Web servers were hacking, session hijacking.

UNIT 3

Cyber Crime and Criminal justice: Concept of Cyber Crime and the IT Act, 2000, Hacking, Teenage Web Vandals, Cyber Fraud and Cheating, Defamation, Harassment and E-mail Abuse, Other IT Act Offences, Monetary Penalties, jurisdiction and Cyber Crimes, Nature of Criminality, Strategies to tackle Cyber Crime and Trends.

UNIT 4

The Indian Evidence Act of 1872 v. Information Technology Act, 2000: Status of Electronic Records as Evidence, Proof and Management of Electronic Records; Relevancy, Admissibility and Probative Value of E-Evidence, Proving Digital Signatures, Proof of Electronic Agreements, Proving Electronic Messages.

UNIT 5

Tools and Methods in Cybercrime: Proxy Servers and Anonymizers, Password Cracking, Key loggers and Spyware, virus and worms, Trojan Horses, Backdoors, DoS and DDoS Attacks , Buffer and Overflow, Attack on Wireless Networks, Phishing : Method of Phishing, Phishing Techniques.

Recommended Books:

1. Principles of Cyber crime, Jonathan Clough Cambridge University Press
2. John R. Vacca, Computer Forensics:Computer Crime Scene Investigation, 2nd Edition, Charles River Media, 2005
3. Cyber Law Simplified, VivekSood, Pub: TMH.
4. Cyber Security by Nina Godbole, SunitBelapure Pub: Wiley-India
5. Information Warfare: Corporate attack and defense in digital world, William Hutchinson, Mathew Warren, Elsevier.
6. Cyber Laws and IT Protection, Harish Chander, Pub:PHI.

RAJIV GANDHI PROUDYOGIKI

VISHWAVIDYALAYA, BHOPAL New Scheme

Based On AICTE Flexible Curricula

Computer Science and Information Technology, V - Semester
Open Elective CSIT- 504 (B) Artificial Intelligence

Course Objectives

1.To present an overview of artificial intelligence (AI) principles and approaches
2.Develop a basic understanding of the building blocks of AI

Course Outcomes: After the completion of this course, the students will be able to:

1.To understand a problem and identify and define the computing requirements appropriate to its solution.
2.To implement and evaluate a computer. based system, process, component, or program to meet desired needs.
3.To apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
4.To apply knowledge representation techniques and problem solving strategies to common AI applications.
5.To design simple software to experiment with various AI concepts and analyze results

UNIT I:

Meaning and definition of artificial intelligence, Production systems, Characteristics of production systems, Study and comparison of breadth first search and depth first search techniques, other Search Techniques like hill Climbing, Best first Search. A* algorithm, AO* algorithms etc, and various types of control strategies.

UNIT II:

Knowledge Representation, Problems in representing knowledge, knowledge representation using propositional and predicate logic, comparison of propositional and predicate logic, Resolution, refutation, deduction, theorem proving, inferencing, monotonic and non-monotonic reasoning.

UNIT III:

Probabilistic reasoning, Baye's theorem, semantic networks, scripts, schemas, frames, conceptual dependency, fuzzy logic, forward and backward reasoning.

UNIT IV:

Game playing techniques like minimax procedure, alpha-beta cut-offs etc, planning, Study of the block world problem in robotics, Introduction to understanding, natural language processing.

UNIT V:

Introduction to learning, Various techniques used in learning, Introduction to neural networks, applications of neural networks, common sense, reasoning, some example of expert systems.

Recommended Books:

1 Rich E and Knight K, “Artificial Intelligence”, TMH, New Delhi.
2 Nelsson N.J., “Principles of Artificial Intelligence”, Springer Verlag, Berlin.

RAJIV GANDHI PROUDYOGIKI

VISHWAVIDYALAYA, BHOPAL New

Scheme Based On AICTE Flexible Curricula

Computer Science & Information Technology, V-Semester

Open Elective CSIT-504 (C) Pattern Recognition

Objectives:

1. Understand the concept of a pattern and the basic approach to the development of pattern recognition and machine intelligence algorithms.
2. Understand the basic methods of feature extraction, feature evaluation, and data mining

Course Outcome: After the completion of this course, the students will be able to:

1. Understand the principle of pattern recognition system and basics of pattern recognition approaches.
2. Introduction and applications of various classification techniques.
3. Understand different paradigms of pattern recognition.
4. Introduction to feature extraction and feature selection.
5. Understand advances in pattern recognition such as PR, SVM, and FCM.

Unit-I

Introduction: Definitions, data sets for Pattern, Application Areas and Examples of pattern recognition, Design principles of pattern recognition system, Classification and clustering, supervised Learning, unsupervised learning and adaptation, Pattern recognition approaches, Decision Boundaries, Decision region , Metric spaces, distances.

UNIT II

Classification: introduction, application of classification, types of classification, decision tree, naïve bayes, logistic regression , support vector machine, random forest, K Nearest Neighbour Classifier and variants, Efficient algorithms for nearest neighbour classification, Different Approaches to Prototype Selection, Combination of Classifiers, Training set, test set, standardization and normalization.

UNIT III

Different Paradigms of Pattern Recognition, Representations of Patterns and Classes, Unsupervised Learning & Clustering: Criterion functions for clustering, Clustering Techniques: Iterative square-error partitional clustering – K means, hierarchical clustering, Cluster validation.

UNIT IV

introduction of feature extraction and feature selection, types of feature extraction , Problem statement and Uses, Algorithms - Branch and bound algorithm, sequential forward / backward selection algorithms, (l,r) algorithm.

UNIT V

Recent advances in Pattern Recognition, Structural PR, SVMs, FCM, Soft computing and Neuro fuzzy techniques, and real-life examples, Histograms rules, Density Estimation, Nearest Neighbor Rule, Fuzzy classification.

Recommended Books:

1. Richard O. Duda, Peter E. Hart and David G. Stork, "Pattern Classification", 2nd Edition, John Wiley, 2006.

2. C. M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2009.

3. S. Theodoridis and K. Koutroumbas, "Pattern Recognition", 4th Edition, academic Press, 2009.

4. Robert Schalkoff, "pattern Recognition: statistical, structural and neural approaches", John Wiley & sons, Inc, 2007.

RAJIV GANDHI PROUDYOGIKI

VISHWAVIDYALAYA, BHOPAL New Scheme

Based On AICTE Flexible Curricula

Computer Science & Information Technology, V-Semester

CSIT-505 Lab (Shell Programming)

Objectives:

1. To provide introduction to UNIX Operating System and its File System
2. To gain an understanding of important aspects related to the SHELL and the process
3. To develop the ability to formulate regular expressions and use them for pattern matching.
4. To provide a comprehensive introduction to SHELL programming, services and utilities.

Course Outcomes: **After Completing the course student should be able to:**

1. Understand the UNIX Architecture, File systems and use of basic Commands.
2. Analyze Vi editor and Networking commands
3. Implement Shell Programming to write shell scripts.
4. Design shell scripts to perform repetitive tasks using while and for loops.
5. Design and implement shell functions.

Unit I

Introduction, Brief history. Unix Components/Architecture. Features of Unix. The UNIX Environment and UNIX Structure, Posix and Single Unix specification. The login prompt. General features of Unix commands/ command structure. Command arguments and options. Understanding basic type command: knowing the type of a command and locating it.

Unit II

The vi editor. Basics.The .exrc file. Different ways of invoking and quitting vi. Different modes of vi. Input mode commands, Command mode commands, the ex mode commands, Navigation commands, Repeat command. Pattern searching. The search and replace command.Wild cards and file name generation. Removing the special meanings of wild cards. Three standard files and redirection. Connecting commands: Pipe.Splitting the output: tee.Command substitution. Basic and Extended regular expressions. The grep, egrep.Typical examples involving different regular expressions.

Unit III

Introduction to Shell programming. Ordinary and environment variables.The .profile. Read and readonly commands. Command line arguments, exit and exit status of a command. Logical operators for conditional execution. The test command and its shortcut.The if, while, for and case control statements. The set and shift commands and handling positional parameters. Trap command. Simple shell program examples.

Unit IV

Regular Expressions:What are regular expressions, Metacharacters, Basic VS Extended regular expressions, Using GREP, Pattern matching, Conditional Statement: IF Advanced usage of IF CASE Statements Writing Interacting Scripts: Displaying user messages, Catching user inputs. Repetitive Tasks: for, while, until, i/o redirection, break & continue, select, shift.

UNIT V

Functions: Introduction, Examples, Overview of Different Shells: Common features, Differing features

Recommended Books:

1. Sumitabha Das., Unix Concepts and Applications., 4 th Edition., Tata McGraw Hill
2. Behrouz A. Forouzan, Richard F. Gilberg : UNIX and Shell Programming Cengage Learning – India Edition. 2009.
3. M.G. Venkatesh Murthy: UNIX & Shell Programming, Pearson Education.
4. Richard Blum , Christine Bresnahan : Linux Command Line and Shell Scripting Bible,2nd Edition , Wiley,2014

List of Practical:

1. To show certain file characteristics in nice columns
2. To show which files from a list do exist
3. To output two different files from one script
4. To ask for user input until valid input is given.
5. To process a list of files, generated by several methods
6. To remember where stdout is going, redirect stdout somewhere else, then restore stdout back to what it was
7. Access each word in each line of a file
8. Using the case statement to implement a menu