Introduction

Content: The B.Sc. (H) Computer Science programme is designed to develop computational thinking, analytical, and problem
solving skills. It covers the core computer science topics like computer systems architecture, data structures, computer networks, operating systems, computer graphics, algorithms, software engineering, database management, theory of computation, artificial intelligence, and information security. The programme builds a base for entry level jobs in information technology and prepares the students for higher studies in the area of Computer Science/Applications.

Learning Outcome based approach to Curriculum Planning

>> Aims of Bachelor's degree programme in (CBCS) B.SC.(HONS.) COMPUTER SCIENCE

Content:
1. Develop theoretical foundations in computer science.
2. Develop expertise in programming skills using high level programming languages.
3. Develop skills to design, implement and document the solutions for computational problems.
4. Develop soft skills to work effectively in a team to solve a problem.
5. Develop the ability to use state of the art technologies.

Graduate Attributes in Subject

>> Disciplinary knowledge

Content:

1. Ability to build (either independently or by joining higher academic program) on of the core computer science concepts learnt in the course.
2. Ability to apply the core computer science concepts to solve the problems in the IT industry.

Graduate Attributes in Subject

>> Problem solving

Content: Graduates are equippped with skills to solve the computational problems at their workplace and for the society.

Graduate Attributes in Subject

>> Cooperation/Team work

Content: Graduates demonstrate competence to use communication skills to participate or lead a team for a new initiative or for solving an existing problem.

Graduate Attributes in Subject

>> Communication Skills

Content: Graduates demonstrate effective communication and presentation skills while interacting with professional peers and in the society.

Graduate Attributes in Subject

>> Scientific reasoning

Content: Given a problem, the graduates will be able to analyze it, suggest solutions, and critically evaluate the solutions proposed by others.

Qualification Description

Content: Demonstrate coherent knowledge and understanding of the logical organization of a digital computer, its components and working. Understanding of the time and space complexities of algorithms designed to solve computational problems.
Demonstrate programming skills in high level language and an ability to learn a new programming language without substantial effort.
Apply knowledge of logical skills to identify and analyse problems and issues, and seek solutions to real-life problems. Eg. creating mobile applications, database applications, educative computer games.

Programme Learning Outcome in course

Content:
1. Ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
2. Ability to design, implement, and evaluate a computer-based system, process, component, or program to solve the given problem.
3. Ability to communicate effectively through oral and written means.
4. Ability to work in a team to achieve a common goal

Combinatorial Optimization
(BHCS15C)
Core Course - (CC) Credit:6

Course Objectives

This course is designed to introduce the fundamentals of combinatorial optimization to the students in terms of both theory and applications, so as to equip them to explore the more advanced areas of convex and non-convex optimizations.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. model problems using linear and integer programs
2. apply polyhedral analysis to develop algorithms for optimization problems
3. use the concept of duality for design of algorithms

Unit 1

Introduction to Combinatorial Optimization Problems, Linear and Integer Programs: LP Formulation, understanding integer programs, computational complexities of IP vs LP, using LP to find optimal or approximate integral solutions, concept of integrality gap.

Unit 2

Theory of Linear Programming and Algorithmic Perspective to Simplex Method: standard vs. equational form, basic feasible solutions, convexity and convex polyhedra, correspondence between vertices and basic feasible solutions, geometry of Simplex algorithm, exception handling (unboundedness, degeneracy, infeasibility), Simplex algorithm, avoiding cycles.

Unit 3

Primal-Dual Algorithms: interpretation of dual, optimality conditions for primal and dual, weak and strong duality, complementary slackness, primal-dual algorithm for the shortest path problem.

Unit 4

Network Flows: linear programming formulations for network flows and bipartite matching, totally uni-modular matrices integral polyhedral.

References

1. Matousek and Gartner, “Understanding and Using Linear Programming”, 2007, Springer.
2. C.H. Papadimitriou and K.Steiglitz, “Combinatorial Optimization: Algorithms and complexity”, 1998, Dover Publications.

Additional Resources:

1. Mokhtar S.Bazaraa, John J. Jarvis and Hanif D. Sherali, “Linear Programming and Network Flows”, 2nd Edition, 2008, Wiley.
2. Bernhard Korte, Jens Vygen, "Combinatorial Optimization", 5th Edition, Springer.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods

- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

optimization problems, linear programming, integer programming, duality, network flow problems

Computer Networks
(BHCS09)
Core Course - (CC) Credit:6

Course Objective

This course covers the concepts of data communication and computer networks. It comprises of the study of the standard models for the layered protocol architecture to communicate between autonomous computers in a network and also the main features and issues of communication protocols for different layers. Topics covered comprise of introduction to OSI and TCP/IP models also.

Course Learning Outcomes

On successful completion of the course, the student will be able to:

1. describe the hardware, software components of a network and their interrelations.
2. compare OSI and TCP/IP network models.
3. describe, analyze and compare different data link, network, and transport layer protocols.
4. design/implement data link and network layer protocols in a simulated networking environment.

Unit 1

Introduction: Types of computer networks, Internet, Intranet, Network topologies, Network classifications.

Unit 2

Network Architecture Models: Layered architecture approach, OSI Reference Model, TCP/IP

Reference Model.

Unit 3

Physical Layer: Analog signal, digital signal, digital modulation techniques (ASK, PSK, QAM), encoding techniques, maximum data rate of a channel, transmission media (guided transmission media, wireless transmission, satellite communication), multiplexing (frequency division multiplexing, time division multiplexing, wavelength division multiplexing).

Unit 4

Data Link MAC Layer: Data link layer services, error-detection and correction techniques, error recovery protocols (stop and wait, go back n, selective repeat), multiple access protocols, (TDMA/FDP, CDMA/FDD/CSMA/CD, CSMA/CA), Datalink and MAC addressing, Ethernet, data link layer switching, point-to-point protocol.

Unit 5

Network layer: Networks and Inter networks, virtual circuits and datagrams, addressing, sub netting, Routing- (Distance vector and link state routing), Network Layer Protocols- (ARP, IPV4, ICMP, IPV6).

UNIT 6

Transport and Application Layer: Process to process Delivery- (client server paradigm, connectionless versus connection oriented service, reliable versus unreliable); User Datagram Protocol, TCP/IP protocol, Flow Control.

UNIT 7

FTP (File Transfer protocol), SMTP (Simple, Mail Transfer Protocol), Telnet and remote login protocol, WWW (World Wide Web), HTTP (Hyper Text Transfer protocol), Uniform Resource Locator, HTML and forms.

References:

1. A.S. Tanenbaum David J. Wethrall, Computer Networks, Pearson publication, 2012
2. Behnouz A. Forouzan, Data Communication and Networking, McGraw-Hill Education, 2017.

Additional References:

1. J.F. Kurose, Keith W. Ross, Computer Networking: A Top-Down Approach, Pearson Education India, 2017.
2. W. Stallings, Data and Computer Communications, 10th Edition, Pearson Education India, 2017.
3. The TCP/IP Guide by Charles M. Kozierok, free online resource, http://www.tcpipguide.com/free/index.htm

Practical

The Practicals shall be based on concepts taught in the course.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

network topologies, OSI model, TCP/IP model, client server model.

Computer System Architecture
(BHCS02)
Core Course - (CC) Credit:6

Course Objective

This course introduces the students to the fundamental concepts of digital computer organization, design and architecture. It aims to develop a basic understanding of the building blocks of the computer system and highlights how these blocks are organized together to architect a digital computer system.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. design Combinational Circuits using basic building blocks. Simplify these circuits using Boolean Algebra and Karnaugh maps. Differentiate between combinational circuits and sequential circuits.
2. represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.
3. determine various stages of instruction cycle and describe interrupts and their handling.
4. explain how CPU communicates with memory and I/O devices.
5. simulate the design of a basic computer using a software tool

Unit 1

Digital Logic Circuits: Logic Gates, truth tables, Boolean Algebra, digital circuits, combinational circuits, sequential circuits, circuit simplification using Karnaugh map, Don’t Care Conditions, flip-flops, characteristic tables

Unit 2

Digital Components: Half Adder, Full Adder, Decoders, Multiplexers, Registers and Memory Units

Unit 3

Data Representation and Basic Computer Arithmetic: Number system,, complements , fixed and floating point representation.Alphanumeric representation. Addition , subtraction,, Booth algo for integer multiplication, Non restoring division of integers.

Unit 4

Basic Computer Organization and Design:Common Bus system, instruction codes, instruction format, instruction set completeness, Sequence Counter, timing and control, instruction cycle, memory reference instructions and their implementation using arithmetic, logical, program control, transfer and input output micro-operations, interrupt cycle.

Unit 5

Central Processing Unit: Micro programmed Control vs Hardwired Control, lower level programming languages, Instruction format, accumulator, general register organization, stack organization, zero-address instructions, one-address instructions, two-address instructions, three-address instructions, Addressing Modes, RISC, CISC architectures, pipelining and parallel processing.

Unit 6

Memory Organization and Input-Output Organization: Input-Output Organization: Peripheral Devices, I/O interface, I/O vs. Memory Bus, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access

Practical

The practicals shall be based on concepts covered in theory.

References

1. M. Mano, Computer System Architecture, Third Edition, Pearson Education

Additional Resources:

1. W. Stallings, Computer Organization and Architecture Designing for Performance, 8th Edition, Prentice Hall of India
2. M. Mano, Digital Design, Pearson Education Asia
3. Linda Null, Julia Lobur, The Essentials of Computer Organization and Architecture, Third Edition, Reprint, Jones and Bartlett

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Combinational and sequential circuits, memory organization, computer organization, CPU design, parallelism.

Data Structures
(BHCS05)
Core Course - (CC) Credit:6

Course Objective

This course aims at developing the ability to use basic data structures like array, stacks, queues, lists, trees and hash tables to solve problems. C++ is chosen as the language to understand implementation of these data structures.

Course Learning Outcomes

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

1. implement and Empirically Analyse linear and non-linear data structures like Arrays, Stacks, Queues, Lists, Trees, Heaps and Hash tables as abstract data structures. (RBT L2/3)
2. write a program, choosing a data structure, best suited for the application at hand. (RBT L3/4)
3. re-write a given program that uses one data structure, using a more appropriate/efficient data structure (RBT L4)
4. write programs using recursion for simple problems. Explain the advantages and disadvantages of recursion.(RBT L2/L3)
5. identify Ethical Dilemmas.

Unit 1

Arrays: single and multi-dimensional arrays, analysis of insert, delete and search operations in arrays (both linear search and binary search), implementing sparse matrices, applications of arrays to sorting: selection sort, insertion sort, bubble sort, comparison of sorting techniques via empirical studies. Introduction to Vectors.

Unit 2

Linked Lists: Singly- linked, doubly-linked and circular lists, analysis of insert, delete and search operations in all the three types, implementing sparse matrices. Introduction to Sequences.

Unit 3

Queues: Array and linked representation of queue, de-queue, comparison of the operations on queues in the two representations. Applications of queues.

Unit 4

Stacks: Array and linked representation of stacks, comparison of the operations on stacks in the two representations, implementing multiple stacks in an array; applications of stacks: prefix, infix and postfix expressions, utility and conversion of these expressions from one to another; applications of stacks to recursion: developing recursive solutions to simple problems, advantages and limitations of recursion

Unit 5

Trees and Heaps: Introduction to tree as a data structure; binary trees, binary search trees, analysis of insert, delete, search operations, recursive and iterative traversals on binary search trees. Height-balanced trees (AVL), B trees, analysis of insert, delete, search operations on AVL and B trees.

Introduction to heap as a data structure. analysis of insert, extract-min/max and delete-min/max operations, applications to priority queues.

Unit 6

Hash Tables: Introduction to hashing, hash tables and hashing functions -insertion, resolving collision by open addressing, deletion, searching and their analysis, properties of a good hash function.

Practical

Programs based on the concepts covered in theory.

References

1. Goodrich, M., Tamassia, R. and Mount D., Data Structures and Algorithms Analysis in C++, second Edition, Wiley., 2011.
2. Adam Drozdek, Data Structures and algorithm in C++, Third Edition, Cengage Learning, 2012.

Additional Resources:

1. Sartaj Sahni, Data Structures, "Algorithms and applications in C++", Second Edition, Universities Press, 2011.
2. Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidyah Langsam, "Data Structures Using C and C++:, Second edition, PHI, 2009.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Arrays and linked lists, stacks, queues, tree, heap, hashing, recursion

Database Management Systems
(BHCS10)
Core Course - (CC) Credit:6

Course Objective

The course introduces the foundations of database management systems focusing on significance of a database, relational data model, schema creation and normalization, transaction processing, indexing, and the relevant data structures (files and B+-trees).

Course Learning Outcomes

On successful completion of the course, students will:

1. describe major components of DBMS and their functions
2. model an application’s data requirements using conceptual modelling tools like ER diagrams and design database schemas based on the conceptual model.
3. write queries in relational algebra / SQL
4. normalize a given database schema to avoid data anomalies and data redundancy.
5. describe the notions of indexes, views, constraints, transactions and their properties (ACID),

Unit 1

Introduction to databases: Characteristics of database approach, data models, database system architecture, data independence and data abstraction.

Unit 2

Data modeling: Entity relationship (ER) modeling: Entity types, relationships, constraints, ER diagrams, EER model

Unit 3

Relation data model: Relational model concepts, relational constraints, relational algebra.

Unit 4

SQL queries: SQL data definition, data types, specifying constraints, Queries for retrieval, insertion, deletion, updation, introduction to views.

Unit 5

Database design: Mapping ER/EER model to relational database, functional dependencies, Lossless decomposition, Normal forms (upto BCNF).

Unit 6

Transaction and data storage:
Introduction to transaction processing: ACID properties, concurrency control; Introduction to indexing structures for files.

Practical

Practicals based on concepts mentioned in theory using relevant software.

References

1. RamezElmasri, ShamkantB. Navathe, Fundamentals of Database Systems (6thEdition), Pearson Education, 2010

Additional Resources:

1. Christopher J.Date, An Introduction to database systems (8thedition), Pearson Education, 2004.
2. AbrahamSilberschatz, Henry F. Korth, Shashank Sudarshan, Database System Concepts (6thEdition), McGrawHill, 2010.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions

- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Entity-Relationship Modeling, Database Design, Transaction Processing, noSQL systems.

Design and Analysis of Algorithms
(BHCS08)
Core Course - (CC) Credit:6

Course Objective

This course is designed to introduce the students to algorithm design and its analysis in terms of efficiency and correctness in general. The course focuses on highlighting difference between various problem solving techniques for efficient algorithm design.

Course Learning Outcomes:

On successful completion of this course, the student will be able to:

1. given an algorithm, identify the problem it solves.
2. write algorithms choosing the best one or a combination of two or more of the algorithm design techniques: Iterative, divide-n-conquer, Greedy, Dynamic Programming using appropriate data structures.
3. write proofs for correctness of algorithms.
4. re-write a given algorithm replacing the (algorithm design) technique used with a more appropriate/efficient (algorithm design) technique.

Unit 1

Algorithm Design Techniques: Iterative technique: Applications to Sorting and Searching (review), their correctness and analysis. Divide and Conquer: Application to Sorting and Searching (review of binary search), merge sort, quick sort, their correctness and analysis.

Dynamic Programming: Application to various problems (for reference; Weighted Interval Scheduling, Sequence Alignment, Knapsack), their correctness and analysis. Greedy Algorithms: Application to various problems, their correctness and analysis.

Unit 2

More on Sorting and Searching: Heapsort, Lower Bounds using decision trees, sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Medians & Order Statistics, complexity analysis and their correctness.

Unit 3

Advanced Analysis Technique: Amortized analysis

Unit 4

Graphs: Graph Algorithms - Breadth First Search, Depth First Search and its Applications.

Practical

Programs based on the concepts covered in theory

References

1. J. Kleinberg, E. Tardos, Algorithm Design, 1st Edition, Pearson Education India, 2013.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms,Third Edition, PHI, 2015
3. Sarabasse, A.V. Gleder, Computer Algorithm – Introduction to Design and Analysis, Pearson, Third edition , 1999

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Brute Force Algorithm, divide and conquer, greedy, dynamic programming approaches, inplace algorithm, best / average / worst case running time of algorithms.

Discrete Structures
(BHCS03)
Core Course - (CC) Credit:6
Course Objective

The course aims to introduce the students to boolean algebra, sets, relations, functions, principles of counting, and growth functions so that these concepts may be used effectively in other courses.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. define mathematical structures (relations, functions, sequences, series, graphs) and use them to model real life situations.
2. understand (trace) and construct simple mathematical proofs using logical arguments.
3. solve class room puzzles based on counting principles.
4. compare functions and relations with respect to their growth for large values of the input.

Unit 1

Introduction: Sets - finite and infinite sets, uncountable infinite sets; functions, relations, properties of binary relations, closure, partial ordering relations; counting - Pigeonhole Principle, permutation and combination; mathematical induction, Principle of Inclusion and Exclusion.

Unit 2

Growth of Functions: asymptotic notations, summation formulas and properties, bounding summations, approximation by integrals.

Unit 3

Recurrences: recurrence relations, generating functions, linear recurrence relations with constant coefficients and their solution, recursion trees, Master Theorem

Unit 4

Graph Theory: basic terminology, models and types, multi-graphs and weighted graphs, graph representation, graph isomorphism, connectivity, Euler and Hamiltonian Paths and Circuits, planar graphs, graph coloring, Trees, basic terminology and properties of Trees, introduction to spanning trees.

Unit 5

Propositional Logic:logical connectives, well-formed formulas, tautologies, equivalences, Inference Theory

Practical

Practicals based on concepts covered in theory.

References

1. C.L. Liu & Mohapatra, Elements of Discrete mathematics, Fourth Edition, 2012, McGraw Hill.
2. Kenneth H Rosen, Discrete Mathematics and Its Applications, Seventh Edition, 2011, McGraw Hill.

Additional Resources:

1. T.H. Cormen, C.E. Leiserson, R. L. Rivest, Introduction to algorithms, 3rd edition, 2009, MIT Press.
2. M. O. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms, 1988, John Wiley and Sons.
3. J. L. Hein, Discrete Structures, Logic, and Computability, 4th Edition, 2015, Jones and Bartlett Learning.
4. D.J. Hunter, Essentials of Discrete Mathematics, 2nd Edition, 2011, Jones and Bartlett Learning.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions

- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

recurrence, trees and graphs, combinatorics, inductive and deductive reasoning, asymptotic complexity.

Information Security
(BHCS14)
Core Course - (CC) Credit:6

Course Objective

The course focuses gives a broad overview of the fundamentals of information security covering topics such as error correction/detection, cryptography, steganography, malwares, The course also touches on the implications of security in Internet of Things (IoT).

Course Learning Outcomes

On successful completion of this course, a student will be able to,

1. identify the major types of threats to information security
2. describe the role of cryptography in security
3. select appropriate error-detection and error-correction methods for an application
4. discuss the strengths and weaknesses of private and public key crypto systems
5. describe malwares and memory exploits

6. discuss the need for security in IoT

Unit 1

Security Concepts, Challenges, Security architecture, Security attacks, security services, security mechanisms

Unit 2

Error detecting/correction - Block Codes, Generator Matrix, Parity Check Matrix, Minimum distance of a Code, Error detection and correction, Standard Array and syndrome decoding, Hamming Codes

Unit 3

Cryptography - Encryption, Decryption, Substitution and Transposition, Confusion and diffusion, Symmetric and Asymmetric encryption, Stream and Block ciphers, DES, cryptanalysis.

Public-key cryptography, Diffie-Hellman key exchange, man-in-the-middle attack

Digital signature, Steganography, Watermarking.

Unit 4

Malicious softwares - Types of malwares (viruses, worms, trojan horse, rootkits, bots), Memory exploits - Buffer overflow, Integer overflow

Unit 5

Security in Internet-of-Things - Security implications, Mobile device security - threats and strategies

References

1. C.P Pfleeger and S.L Pfleeger; Security in Computing
2. William Stallings; Cryptography and network security, Pearson,2018,7th Ed.
3. Shu Lin, D.J Costello; Error Control Coding: Fundamentals and applications

Additional Resources:

1. E R Berlekamp,. McGraw Hill Book Company,1986.AlgebraicCoding Theory
2. William Stallings; Network security, essentials, Pearson,2018,6th Ed.

3. Whitman M.E.,Mattrod HJ, Principle of Information Security, Thomson,2007

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Security mechanisms, private and public key cryptography, malware detection, security in IoT.

Internet Technologies
(BHCS13)
Core Course - (CC) Credit:6

Course Objective

This course introduces the protocols used in Internet, its architecture, and security aspect of Internet. Student will have an insight that how a search engine works and web crawls.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. describe Internet, its architecture, services and protocol.
2. implement a simple search engine.
3. implement a web crawler.

4. use javascript technologies to make a website highly responsive, more efficient and user friendly

Unit 1

Network address translation, Subnet Masking, Difference between Intranet and Internet, Working of Internet,

Dynamic and Static Routing, Domain Name Server , networking tools - ipconfig, ping, netstat, traceroute

Unit 2

Introduction to Internet Protocols - HTTP, HTTPS, FTP, SMTP, IMAP, POP3, VoIP

Unit 3

Web Servers: Introduction, Working, Configuring, Hosting and Managing a Web server,

Proxy Servers: Introduction, Working, Type of Proxies, setting up and managing a proxy server

Client-side Technologies, Server-side Technologies and hybrid technologies

Unit 4

Javascript, jQuery, JSON, NODE.js, BOOTSTRAP

Introduction to forums, blogging, portfolio

Developing a responsive website

Combining Web Applications and Mobile Applications

Unit 5

Search Engines - components, working, optimization

Crawling, BOTS

Unit 6

Introduction to cookies and sessions.

Introduction to e-commerce websites and e-carts.

Practical

Pre-requisites for course: Programming, Computer Networks,Web-Designing(HTML, CSS, Basic Javascript)

1. Use of networking tools - ping, ipconfig, netstat, traceroute
2. Configuring your own web-server.
3. Building an interactive website using jquery, JSON, NODE.js, BOOTSTRAP

References

1. Ivan Bayross, “Web enabled commercial application development using HTML, JavaScript, DHTML and PHP”, BPB Publication 4th Edition, 2013.
2. The Internet Book: Everything you need to know about computer networking and how the Internet works, Paperback, 4th Edition 2007. ISBN 0-13-233553-0

Additional Resources:

https://www.tutorialspoint.com/seo/

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Internet, networks, JSON, AJAX, JQUERY, web application

Operating system
(BHCS06)
Core Course - (CC) Credit:6

Course Objective

The course introduces the students to different types of operating systems. Operating system modules such as memory management, process management and file management are covered in detail.

Course Learning Outcomes

On successful completion of the course, the students will be able to:

1. implement multiprogramming, multithreading concepts for a small operating system.
2. create, delete, and synchronize processes for a small operating system.
3. implement simple memory management techniques.
4. implement CPU and disk scheduling algorithms.
5. use services of modern operating system efficiently
6. implement a basic file system.

Unit 1

Introduction : Operating systems (OS) definition, Multiprogramming and Time Sharing operating systems, real time OS, Multiprocessor operating systems, Multicore operating systems, Various computing environments.

Unit 2

Operating System Structures: Operating Systems services, System calls and System programs, Operating system architecture (Micro Kernel, client server) operating

Unit 3

Process Management: Process concept, Operation on processes, Multi-threaded processes and models, Multicore systems, Process scheduling algorithms, Process synchronization. The Critical-section problem and deadlock characterization, deadlock handling.

Unit 4

Memory Management: Physical and Logical address space; Memory allocation strategies - Fixed and Variable Partitions,Paging, Segmentation, Demand Paging and virtual memory, Page Replacement algorithm.

Unit 5

File and I/O Management: Directory structure, File access methods, Disk scheduling algorithms.

Practical

References

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 9th Edition, John

1. Wiley Publications.
2. William Stallings, Operating Systems: Internals and Design Principles, 9th edition, Pearson Education, 2018.
3. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
4. D. M. Dhamdhere, Operating Systems: A Concept-based Approach, 2E, Tata McGraw-Hill Education, 2006
5. Kernighan, Brian W., and Rob Pike. The Unix programming environment. Vol. 270. Englewood Cliffs, NJ: Prentice-Hall, 1984.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions

- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Types of operating systems, memory management, process management, file and I/O management

---

Probability & Statistics for Computer Science
(BHCS07)
Core Course - (CC) Credit:6

Course Objectives:

This course introduces the students to the fundamental concepts and topics of probability and statistics, whose knowledge is important in other computer science courses. The course aims to build the foundation for some of the core courses in later semesters .

Course Learning Outcomes

On successful completion of the course, the students will be able to:

1. use probability theory to evaluate the probability of real world events.
2. apply probability theory using Bayes' rule.
3. describe discrete and continuous probability distribution functions.
4. use descriptive statistical methods to summarize, and display data in a meaningful way.
5. use principle of least squares to build regression model.

Unit 1

Probability : Sample Space and Events, Probability axioms, Conditional Probability, Bayes' law,

Independent events.

Unit 2

Random Variables : Discrete and continuous random variable, Bernoulli trials, Binomial random variable, Geometric random variable, Poisson random variable; Uniform distribution, Exponential distribution, Normal distribution, Pareto distribution, Applications

Unit 3

Expected Value, Conditional expectation ; Law of large numbers, Central Limit Theorem, Joint probability distribution, Covariance and Correlation

Unit 4

Statistics: Introduction to Statistics, Descriptive and Inferential Statistics, Population and Samples, Frequency tables, Relative frequency tables and graphs, Grouped data, Histograms, Ogives, Mean, Median, Mode, Standard Deviation, Variance, Percentiles, Coefficient of variation, Skewness, Kurtosis; Box plot, Scatter diagram.

Unit 5

Simple Linear Regression, Principle of least squares and Fitting of polynomials and Exponential curves

References:

1. From Algorithms to Z-Scores: Probabilistic and Statistical Modeling in Computer Science, Norm Matloff, University of California, Davis
2. Linear Algebra and Probability for Computer Science Applications, Ernest Davis, CRC Press

Additional Resources:

1. Sheldon M. Ross: Introduction to Probability and Statistics for Engineers and Scientists, fourth edition
2. Probability and Statistics for Computer Scientists, Michael Baron, CRC Press
3. Kishor S. Trivedi. Probability and Statistics with Reliability, Queuing, and Computer Science Applications, John Wiley and Sons, New York, second edition

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Probability, Random variable, Probability distribution, Descriptive statistics, Correlation and Regression.

Programming Fundamentals using Python
(BHCS01)
Core Course - (CC) Credit:6

Course Objective

This course is designed to introduce student to the basics of programming using Python. The course covers the topics essential for developing well documented modular programs using different instructions and built-in data structures available in Python.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. solve simple computational problems.
2. select a suitable programming construct and inbuilt data structure for a situation.
3. use built-in strings, lists, sets, tuples and dictionary in an application.
4.

define classes and use them in application programs.

5. use files for I/O operations.

Unit 1

Introduction to Programming using Python : Structure of a Python Program, Functions, Interpreter shell, Indentation. Identifiers and keywords, Literals, Strings, Basic operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment Operator, Bit wise operator).

Unit 2

Building blocks of Python: Standard libraries in Python, notion of class, object and method.

Unit 3

Creating Python Programs : Input and Output Statements, Control statements:-branching, looping, Exit function, break, continue and pass, mutable and immutable structures. Testing and Debugging a program

Unit 4

Built-in data structures: Strings, lists, Sets, Tuples and Dictionary and associated operations. Basic searching and sorting methods using iteration and recursion.

Unit 5

Visualization using 2D and 3D graphics: Visualization using graphical objects like Point, Line, Histogram, Sine and Cosine Curve, 3D objects

Unit 6

Exception Handling and File Handling: Reading and writing text and structured files, Errors and Exceptions.

Practical

Programs based on concepts covered in theory

References

1. Sheetal Taneja, Naveen Kumar. Python Programming- A modular Approach. Pearson, 2017.

2. Allen B. Downey, Think Python–How to think like a Computer Scientist. O'Reilly, Second

Edition, 2015.

Additional Resources:

1. John V. Guttag, Introduction to computation and programming using Python. MIT Press, 2016.
2. Y. Daniel Liang, Introduction to programming using Python. Pearson, 2013.
3. R. G. Dromey, How to Solve it by Computer, Pearson, 2006.
4. Martin C. Brown, The Complete Reference: Python, McGraw Hill Education.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Python Program, Control structure, Decision making, Functions, Strings, Lists, Dictionary, Object Oriented Programming

Programming using C++
(BHCS04)
Core Course - (CC) Credit:6

Course Objective

This course is designed to develop structured as well as object-oriented programming skills using C++ programming language. The course not only focuses on basic C++ constructs but also covers object-oriented programming features in-depth, namely Encapsulation, Abstraction, Inheritance and Polymorphism for writing efficient codes.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. explain significance of object oriented paradigm
2. solve programming problems using object oriented features.
3. handle external files as well as exceptions.
4. reuse classes to create new classes.
5. handle exceptions in programs.

Unit 1

Introduction to C++: Overview of Procedural Programming and Object-Oriented Programming, Using main () function, Header Files, Compiling and Executing Simple Programs in C++.

Unit 2

Programming Fundamentals: Data types, Variables, Operators, Expressions, Arrays, Keywords, Naming Convention, Decision making constructs (if, switch), Looping (for, while, do…while), Type Casting, Input-output statements, Functions, Command Line Arguments/Parameters.

Unit 3

Object Oriented Programming: Overview of Abstraction, Encapsulation, Inheritance, and Polymorphism. Creating Classes and objects, Modifiers and Access Control, Constructors, Implementation of Inheritance (Single and multilevel), Implementation of Polymorphism (Function Overloading and Operator Overloading, Function Overriding).

Unit 4

Pointers and References: Static and dynamic memory allocation, Pointer and Reference Variables, Pointers vs. References, Implementing Runtime polymorphism using pointers and references.

Unit 5

Exception and File Handling: Using try, catch, throw, throws and finally; Nested try, creating user defined exceptions, File I/O Basics, File Operations.

Practical

Programs based on the concepts covered in theory.

References

1. Forouzan & Gilbert, Computer Science: A Structured Approach using C++, Cengage Learning.

2. E Balaguruswamy, Object Oriented Programming with C++, 7th edition, McGraw-Hill Education, 2017.

Additional Resources:

1. Stephen Prata, C++ Primer Plus, 6th Edition, Pearson India, 2015

2. Yashavant P. Kanetkar, Let us C++, 2nd Edition, BPB Publishers, 2015

3. Herbert Schildt, C++: The Complete Reference, 4th Edition, McGraw Hill, 2003

4. Bjarne Stroustrup, The C++ Programming Language, 4th Edition, Pearson Education, 2013

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Procedural and Object Oriented programming, abstraction, inheritance, polymorphism, pointers, exception and file handling.

Software Engineering
(BHCS11)
Core Course - (CC) Credit:6

Course Objective

The course introduces fundamental Software Engineering approaches and techniques for software development. The students also develop a case study using appropriate software model.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. analyse and model customer's requirements and model its software design.
2. use suitable software model for the problem at hand.
3. estimate cost and efforts required in building software.
4. analyse and compute impact of various risks involved in software development.
5. design and build test cases, and to perform software testing.

Unit 1

Introduction: Software Engineering - A Layered Approach; Software Process – Process Framework, Umbrella Activities; Process Models – Waterfall Model, Incremental Model, and Evolutionary process Model (Prototyping, Spiral Model); Introduction to Agile – Agility Principles, Agile Model –
Scrum.
[12 Hours]

Unit 2

Software Requirements Analysis and Specifications: Use Case Approach, Software Requirement Specification Document, Flow oriented Modeling, Data Flow Modeling, Sequence Diagrams [8 Hours]

Unit 3

Design Modeling - Translating the Requirements model into the Design Model, The Design Process, Design Concepts - Abstraction, Modularity and Functional Independence; Architectural Mapping using Data Flow. [8 Hours]

Unit 4

Software Metrics and Project Estimations: Function based Metrics, Software Measurement, Metrics for Software Quality; Software Project Estimation (FP based estimations, COCOMO II Model); Project Scheduling (Timeline charts, tracking the schedule). [8 Hours]

Unit 5

Quality Control and Risk Management- Quality Control and Quality Assurance, Software Process Assessment and Improvement Capability Maturity Model Integration (CMMI); Software Risks, Risk Identification, Risk Projection and Risk Refinement, Risk Mitigation, Monitoring and Management.
[8 Hours]

Unit 6

Software Testing: Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing; Black-Box and White Box Testing, Basis Path Testing. [12 Hours]

Practical

Project based on a Case-study that includes the following concepts:

Problem Statement, Process Model, Software Requirements, Project Estimation, Project Scheduling, Risk Management, Software Design, Coding of a small module and Testing.

References

1. RS Pressman, BR Maxim, Software Engineering: A Practitioner's Approach, 8th Ed, McGraw-Hill, 2015.
2. P Jalote, An Integrated Approach to Software Engineering, 3rd Ed, Narosa Publishing House, 2005.

Additional Resources:

1. The Definitive Guide to Scrum: The Rules of the Game, Ken Schwaber, Jeff Sutherland, July 2016. [https://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf]
2. I Sommerville, Software Engineering (9th edition), Addison Wesley, 2011.
3. K.K Aggarwal, Yogesh Singh, Software Engineering (3rd ed.), New Age International Publishers, 2007

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods

- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations, projects.

Keywords

Software models, requirement analysis, software design and testing, software risks and costs.

Theory of Computation
(BHCS12)
Core Course - (CC) Credit:6

Course Objective

This course introduces formal models of computation, namely, finite automaton, pushdown automaton, and Turing machine; and their relationships with formal languages. Students will also learn about the limitations of computing machines.

Course Learning Outcomes

On successful completion of the course, a student will be able to:

1. design a finite automaton, pushdown automaton or a Turing machine for a problem at hand.
2. apply pumping lemma to prove that a language is non-regular/non-context-free.
3. describe limitations of a computing machine.

Unit 1

Languages: Alphabets, string, language, basic operations on language, concatenation, union, Kleene star.

Unit 2

Regular Expressions and Finite Automata : Regular expressions, Deterministic finite automata (DFA).

Unit 3

Regular Languages: Non-deterministic Finite Automata (NFA), relationship between NFA and DFA, Transition Graphs (TG), properties of regular languages, the relationship between regular languages and finite automata, Kleene's Theorem.

Unit 4

Non-Regular Languages and Context Free Grammars: Pumping lemma for regular grammars, Context-Free Grammars (CFG),

Unit 5

Context-Free Languages (CFL) and PDA: Deterministic and non-deterministic Pushdown Automata (PDA), parse trees, leftmost derivation, pumping lemma for CFL, properties of CFL.

Unit 6

Turing Machines and Models of Computations: Turing machine as a model of computation, configuration of simple Turing machine, Church Turing Thesis, Universal Turing Machine, decidability, halting problem.

Practical

Tutorials based on topics covered in theory.

References

1. Daniel I.A. Cohen, Introduction to Computer Theory, Wiley India Pvt. Ltd., 2nd Edition, 2011.
2. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, 2nd Edition, Prentice Hall of India (PHI), 2002.

Additional Resources

1. P. Linz, An Introduction to Formal Languages and Automata, 6/e, Jones and Bartlett, 2016.

2. G.L. Gopalkrishnan, Automata and Computability: A programmer's perspective, CRC Press, 2019.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Regular expressions and languages, finite automata, context free grammar and languages, pushdown automata, Turing machine.

Advanced Data Structures and Algorithms (BHCS17C)Discipline Specific Elective - (DSE) Credit:6

Course Objective

This course focuses on the study of advanced data structures and algorithms for solving problems efficiently and their theoretical behavior. The course also includes study of network flow algorithms, NP completeness and backtracking.

Course Learning Outcomes

On successful completion of this course, the student will be able to:

1. impliment and empirically analize advanced data-structures like tries, suffix trees.
2. apply amortised analysis.
3. develop more sophisticated algorithms using techniques like divide and conquer, dynamic programming, greedy strategy, and augmentation
4. prove that certain problems are too hard to admit fast solutions.
5. develop algorithms using backtracking for the hard problems.

Unit 1

Advanced Data Structures : Skip Lists, Red-Black trees, Splay Trees, Mergeable heaps (Fibonacci heaps), DS for sets - Union-Find

Data Structure, Dynamic Tables, Dictionaries, Data structures for strings - Tries, Suffix trees.

Unit 2

Divide and Conquer : Counting Inversions, Closest pair of points, Integer Multiplication,

Unit 3

Greedy Algorithm: Interval Scheduling, Huffman Code, Correctness and Analysis,

Unit 4

Dynamic Programming: Segmented Least Squares, Shortest Paths, Negative Cycles in Graphs

Unit 5

Network Flows: Max-flow problem, Ford Fulkerson Algorithm, Maximum flows and Minimum Cuts in a network, Bipartite Matching.

Unit 6

NP Completeness : Polynomial time reductions, Efficient Certification and Definition of NP, NP Complete problems, Sequencing problems, Partitioning problems, co-NP and asymmetry of NP.

Backtracking: Constructing All Subsets, Constructing All Permutations, Constructing All Paths in a Graph.

Practical

Tutorials based on the concepts covered in class.

References

1. J. Kleinberg and E.Tardos, “Algorithm Design”, 1st Edition 2013., Pearson Education India
2. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, “Introduction to Algorithms”, 3rd Edition, 2010, Prentice-Hall of India Learning Pvt. Ltd.
3. S. Basse, A.V. Gleder, Computer Algorithm – Introduction to Design and Analysis, Pearson, 3rd edition , 1999
4. Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, "Algorithms", 1st Edition, 2017, Tata McGraw Hill.
5. Steven S Skiena, The Algorithm Design Manual, Springer-Verlag London, Second edition, 2008

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Algorithms, Analysis, Network Flows, NP Completeness.

Artificial Intelligence (BHCS15A)Discipline Specific Elective - (DSE) Credit:6

Course Objective

This course introduces the basic concepts and techniques of Artificial Intelligence (AI). The course aims to introduce intelligent agents and reasoning, heuristic search techniques, game playing, knowledge representation, reasoning with uncertain knowledge.

Course Learning Outcomes

On successful completion of this course, students will be able to:

1. identify problems that are amenable to solution by specific AI methods
2. represent knowledge in Prolog and write code for drawing inferences.
3. identify appropriate AI technique for the problem at hand
4. compare strengths and weaknesses of different artificial Intelligence techniques.

Unit 1

Introduction: Introduction to artificial intelligence, background and applications, Turing test, rational agents, intelligent agents, structure, behavior and environment of intelligent agents.

Unit 2

Knowledge Representation: Propositional logic, first order predicate logic, resolution principle, unification, semantic nets, conceptual dependencies, frames, scripts, production rules, conceptual graphs.

Unit 3

Reasoning with Uncertain Knowledge : Uncertainty, non-monotonic reasoning, truth maintenance systems, default reasoning and closed world assumption, Introduction to probabilistic reasoning, Bayesian probabilistic inference, introduction to fuzzy sets and fuzzy logic, reasoning using fuzzy logic.

Unit 4

Problem Solving and Searching Techniques: Problem characteristics, production systems, control strategies, breadth first search, depth first search, hill climbing and its variations, heuristics search techniques: best first search, A* algorithm, constraint satisfaction problem, means-end analysis.

Unit 5

Game Playing: introduction to game playing, min-max and alpha-beta pruning algorithms.

Prolog Programming: Introduction to Programming in Logic (PROLOG), Lists, Operators, basic Input and Output.

Unit 6

Understanding Natural Languages: Overview of linguistics, Chomsky hierarchy of grammars, parsing techniques.

Practical

Programs based on concepts covered in theory classes.

References

1. Elaine Rich and Kevin Knight, Artificial Intelligence – Tata McGraw Hill, 3rd edition, 2012
2. Stuart J. Russell and Peter Norvig, Artificial Intelligence - A Modern Approach, Pearson, 3rd edition, 2015.

Additional Resources:

1. DAN.W. Patterson, Introduction to A.I and Expert Systems, 1st Edition, Pearson, 2015.
2. Ivan Bratko, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson Education, 4th edition, 2011.
3. Saroj Kaushik, Artificial Intelligence, Cengage Learning India, 2011.
4. W.F. Clocksin and Mellish, Programming in PROLOG, 5th edition, Springer, 2003.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Artificial Intelligence, Problem Solving, Knowledge Representation, Reasoning, Uncertainty, Natural Language Processing

Computer Graphics (BHCS17A)Discipline Specific Elective - (DSE) Credit:6

Course Objective

This course introduces fundamental concepts of Computer Graphics with focus on modelling, rendering and interaction aspects of computer graphics. The course emphasizes the basic principles needed to design, use and understand computer graphics system.

Course Learning Outcomes

On successful completion of the course, students will be able to :

1. describe Standard raster and vector scan devices as well as Graphical Input and output devices
2. implement algorithms for drawing basic primitives such as line,circle and ellipse.
3. implement algorithms for line clipping and polygon clipping and filling.
4. implement a 3D object representation scheme and carryout 2D and 3D Transformation, 3D projections
5. implement visible surface determination algorithms, Illumination models and surface rendering methods, color models
6. implement a simple computer animation algorithm

Unit 1

Introduction: Introduction to Graphics systems, Basic elements of Computer graphics, Applications of computer graphics. Architecture of Raster and Random scan display devices, input/output devices.

Unit 2

Drawing and clipping primitives: Raster scan line, circle and ellipse drawing algorithms, Polygon filling, line clipping and polygon clipping algorithms

Unit 3

Transformation and Viewing: 2D and 3D Geometric Transformations, 2D and 3D Viewing Transformations (Projections- Parallel and Perspective), Vanishing points.

Unit 4

Geometric Modeling: Polygon Mesh Representation, Cubic Polynomial curves (Hermite and Bezier).

Unit 5

Visible Surface determination and Surface Rendering: Z-buffer algorithm, List-priority algorithm and area subdivision algorithm for visible surface determination. Illumination and shading models , RGB color model and Basics of Computer Animation.

Practical

Programs based on concepts covered in theory classes.

References

1. James D. Foley , Andries van Dam, Steven K. Feiner , John F. Hughes, Computer Graphics: Principles and Practice in C (2nd Edition) , Addison-Wesley Professional
2. D.Hearn, Baker: Computer Graphics, (2nd Edition) Prentice Hall of India, 2008.

Additional Resources:

1. D.F.Rogers, Procedural Elements for Computer Graphics, McGraw Hill 1997.
2. S. Marschner, P.Shirley Fundamentals of Computer Graphics, (Fourth Edition), CRC Press,2017.
3. Samit Bhattacharya, Computer Graphics , Oxford University Press.
4. D.F. Rogers, Adams Mathematical Elements for Computer Graphics, McGraw Hill 2nd edition 1989.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions

- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Computer Graphics, Modelling, Rendering, Transformation and viewing

Data Analysis and Visualization
(BHCS16C)
Discipline Specific Elective - (DSE) Credit:6

Course Objective

This course introduces students to data analysis and visualization in the field of exploratory data science using Python.

Course Learning Outcomes

On successful completion of the course, the students will be able to :

1. use data analysis tools in the pandas library.
2. load, clean, transform, merge and reshape data.
3. create informative visualization and summarize data sets.
4. analyze and manipulate time series data.
5. solve real world data analysis problems.

Unit 1

Introduction: Introduction to Data Science, ExploratoryData Analysis and Data Science Process. Motivation for using Python for Data Analysis, Introduction of Python shell iPython and Jupyter Notebook.

Essential Python Libraries: NumPy, pandas, matplotlib, SciPy, scikit-learn, statsmodels

Unit 2

Getting Started with Pandas: Arrays and vectorized conputation.

Introduction to pandas Data Structures, Essential Functionality, Summarizing and Computing Descriptive Statistics.

Data Loading, Storage and File Formats.

Reading and Writing Data in Text Format, Web Scraping, Binary Data Formats, Interacting with Web APIs, Interacting with Databases

Data Cleaning and Preparation.

Handling Missing Data, Data Transformation, String Manipulation

Unit 3

Data Wrangling:Hierarchical Indexing, Combining and Merging Data Sets Reshaping and Pivoting.

Data Visualization matplotlib: Basics of matplotlib, plotting with pandas and seaborn, other python visualization tools

Unit 4

Data Aggregation and Group operations: Group by Mechanics, Data aggregation, General split-apply-combine, Pivot tables and cross tabulation

Time Series Data Analysis: Date and Time Data Types and Tools, Time series Basics, date Ranges, Frequencies and Shifting, Time Zone Handling, Periods and Periods Arithmetic, Resampling and Frequency conversion, Moving Window Functions.

Unit 5

Advanced Pandas: Categorical Data, Advanced GroupBy Use, Techniques for Method Chaining

Practical

Practicals based on Analysis and visualization of public datasets and interpretation of the analysis.

Data Analysis Case Studies

References

1. Cathy O'Neil, Rachel Schutt, O'Reilly, 'Doing Data Science: Straight Talk from the Frontline', 2014
2. Wes McKinney, O'Reilly, 'Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython', Second Edition 2017.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Data Analysis, data wrangling, data visualization, data cleaning, data preparation

Data Mining
(BHCS17B)
Discipline Specific Elective - (DSE) Credit:6

Course Objective

This course introduces data mining techniques and enables students to apply these techniques on real-life datasets. The course focuses on three main data mining techniques: Classification, Clustering and Association Rule Mining tasks.

Course Learning Outcomes

On successful completion of the course, students will be able to do following:

1. pre-process the data, and perform cleaning and transformation.
2. apply suitable classification algorithm to train the classifier and evaluate its performance.
3. apply appropriate clustering algorithm to cluster data and evaluate clustering quality
4. use association rule mining algorithms and generate frequent item-sets and association rules

Unit 1

Introduction to Data Mining - Applications of data mining, data mining tasks, motivation and challenges, types of data attributes and measurements, data quality.

Data Pre-processing - aggregation, sampling, dimensionality reduction, Feature Subset Selection, Feature Creation, Discretization and Binarization, Variable Transformation.

Unit 2

Classification: Basic Concepts, Decision Tree Classifier: Decision tree algorithm, attribute selection measures, Nearest Neighbour Classifier, Bayes Theorem and Naive Bayes Classifier, Model Evaluation: Holdout Method, Random Sub Sampling, Cross-Validation, evaluation metrics, confusion matrix.

Unit 3

Association rule mining : Transaction data-set, Frequent Itemset, Support measure, Apriori Principle, Apriori Algorithm, Computational Complexity, Rule Generation, Confidence of association rule.

Unit 4

Cluster Analysis: Basic Concepts, Different Types of Clustering Methods, Different Types of Clusters, K-means: The Basic K-means Algorithm, Strengths and Weaknesses of K-means algorithm, Agglomerative Hierarchical Clustering: Basic Algorithm, Proximity between clusters, DBSCAN: The DBSCAN Algorithm, Strengths and Weaknesses.

Practical

Practicals based on concepts covered in theory classes.

References

1. Introduction to Data Mining, 2nd Edition, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson Education.
2. Data Mining: Concepts and Techniques, 3nd Edition, Jiawei Han and Micheline Kamber

Additional References

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

data mining, classifiers, data pre-processing, metrics.

Deep Learning
(BHCS18B)
Discipline Specific Elective - (DSE) Credit:6

Course Objective

The objective of this course is to introduce students to deep learning algorithms and their applications in order to solve real problems.

Course Learning Outcomes

On successful completion of this course, the student will be able to:

1. describe the feedforward and deep networks.
2. design single and multi-layer feed-forward deep networks and tune various hyper-parameters.
3. implement deep neural networks to solve a problem
4. analyse performance of deep networks.

Unit 1

Introduction: Historical context and motivation for deep learning; basic supervised classification task, optimizing logistic classifier using gradient descent, stochastic gradient descent, momentum, and adaptive sub-gradient method.

Unit 2

Neural Networks: Feedforward neural networks, deep networks, regularizing a deep network, model exploration, and hyperparameter tuning.

Unit 3

Convolution Neural Networks: Introduction to convolution neural networks: stacking, striding and pooling, applications like image, and text classification.

Unit 4

Sequence Modeling: Recurrent Nets: Unfolding computational graphs, recurrent neural networks (RNNs), bidirectional RNNs, encoder-decoder sequence to sequence architectures, deep recurrent networks, LSTM networks.

Unit 5

Autoencoders: Undercomplete autoencoders, regularized autoencoders, sparse autoencoders, denoising autoencoders, representational power, layer, size, and depth of autoencoders, stochastic encoders and decoders.

Unit 6

Structuring Machine Learning Projects: Orthogonalization, evaluation metrics, train/dev/test distributions, size of the dev and test sets, cleaning up incorrectly labeled data, bias and variance with mismatched data distributions, transfer learning, multi-task learning.

Practical

Practicals based on concepts covered in theory.

References:

1. N. Bunduma, Fundamentals of Deep Learning, Oreilly Books, 2017.
2. Jeff Heaton, Deep Learning and Neural Networks, Heaton Research Inc, 2015.

Additional references:

1. Ian Goodfellow, Deep Learning, MIT Press, 2016.
2. Mindy L Hall, Deep Learning, VDM Verlag, 2011.
3. Li Deng (Author), Dong Yu, Deep Learning: Methods and Applications (Foundations and Trends in Signal Processing), Now Publishers Inc, 2009.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Convolution Neural Networks, Recurrent nets, autoencoders

=""">

Digital Image Processing
(BHCS16A)
Discipline Specific Elective - (DSE) Credit:6

Course Objective

This course introduces students to the fundamentals of digital image processing, and various image transforms, image restoration techniques, image compression and segmentation used in digital image processing.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. describe the roles of image processing systems in a variety of applications;
2. write programs to read/write and manipulate images: enhancement, segmentation, and compression, spatial filtering.
3. develop Fourier transform for image processing in frequency domain.
4. evaluate the methodologies for image segmentation, restoration

Unit 1

Introduction: Digital Image Fundamentals: Brightness, Adaptation and Discrimination, Light and Electromagnetic Spectrum, Image Sampling and Quantization, Some Basic Relationships Between Pixels Types of images.

Unit 2

Spatial Domain Filtering: Some Basic Intensity Transformation Functions, Histogram Equalization, Spatial Correlation and Convolution, Smoothening Spatial Filters: Low pass filters, Order Statistics filters; Sharpening Spatial Filters: Laplacian filter

Unit 3

Filtering in Frequency Domain: The Discrete Fourier Transformation(DFT), Frequency Domain Filtering:Ideal and Butterworth Low pass and High pass filters, DCT Transform (1D, 2D).

Unit 4

Image Restoration: Image Degradation/Restoration Process, Noise models, Noise Restoration Filters

Image Compression: Fundamentals of Image Compression, Huffman Coding, Run Length Coding, JPEG.

Unit 5

Morphological Image Processing: Erosion, Dilation, Opening, Closing, Hit-or-Miss Transformation, Basic Morphological Algorithms.

Unit 6

Image Segmentation: Point, Line and Edge Detection, Thresholding, Region Based Segmentation.

Practicals

Based on concepts covered in the course using MATLAB.

References

1. R C Gonzalez, R E Woods, Digital Image Processing, 3rd Edition, Pearson Education
2. A K Jain, Fundamentals of Digital Image Processing, Prentice Hall of India
3. K R Castleman, Digital Image Processing, Pearson Education
4. Schalkoff, Digital Image Processing and Computer Vision, John Wiley and Sons
5. R C Gonzalez, R E Woods, Steven Eddins, Digital Image Processing using MATLAB, Pearson Education, Inc.,2004

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

image transform, image restoration, image processing, image segmentation.

Machine Learning
(BHCS18A)
Discipline Specific Elective - (DSE) Credit:6

Course Objective

The course aims at introducing the basic concepts and techniques of machine learning so that a student can apply machine learning techniques to a problem at hand.

Course Learning Outcomes

On successful completion of this course, the student will be able to:

1. differentiate between supervised and unsupervised learning tasks.
2. differentiate between linear and non-linear classifiers.
3. describe theoretical basis of SVM
4. Implement various machine learning algorithms learnt in the course.

Unit 1

Introduction: Basic definitions, Hypothesis space and inductive bias, Bayes optimal classifier and Bayes error, Occam's razor, Curse of dimensionality, dimensionality reduction, feature scaling, feature selection methods.

Unit 2

Regression: Linear regression with one variable, linear regression with multiple variables, gradient descent, logistic regression, over-fitting, regularization. performance evaluation metrics, validation methods.

Unit 3

Classification: Decision trees, Naive Bayes classifier, k-nearest neighbor classifier, perceptron, multilayer perceptron, neural networks, back-propagation algorithm, Support Vector Machine (SVM), Kernel functions.

Unit 4

Clustering: Approaches for clustering, distance metrics, K-means clustering, expectation maximization, hierarchical clustering, performance evaluation metrics, validation methods.

Practical

Experiments based of concepts covered in the course.

References

1. Peter Flach, Machine Learning: The Art And Science Of Algorithms That Make Sense Of Data, Cambridge University Press, 2015
2. Tom M. Mitchell, Machine Learning, McGraw Hill Education, 2017

Additional References:

1. Christopher, M. Bishop, Pattern Recognition And Machine Learning, Springer-Verlag New York, 2016.
2. Simon O. Haykin: Neural Networks and Learning Machines, PHI, 3rd Edition, 2010.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

machine learning, unsupervised learning, supervised learning, support vector machines, neural networks, classification, clusterning,

---

Microprocessors
(BHCS16B)
Discipline Specific Elective - (DSE) Credit:6

Course Objective

This course introduces internal architecture, programming model of Intel Microprocessors (8086 - Pentium) and assembly language programming using an assembler. Students will also learn interfacing of memory and I/O devices with microprocessor.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. describe the internal architecture of Intel microprocessors
2. define and implement interfaces between the microprocessor and the devices.
3. write assembly language programs

Unit 1

Microprocessor architecture: Internal architecture, Programming Model, Addressing modes, Data movement instructions

Unit 2

Microprocessor programming: Register Organization, instruction formats, Program control instructions, assembly language

Unit 3

Interfacing: Bus timings, Memory address decoding, cache memory and cache controllers, I/O interface, keyboard, timer, interrupt controller, DMA controller, video controllers, communication interfaces.

Practical

The Practicals shall be based on concepts taught in the course.

References

1. Barry B. Brey: The Intel Microprocessors: Architecture, Programming and Interfacing. Pearson Education, Eighth Edition. 2009.
2. Walter A Triebel, Avtar Singh: The 8088 and 8086 Microprocessors Programming, Interfacing, Software

---

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Microprocessor architecture, microprocessor programming, interfacing,

---

Systems Programming
(BHCS15B)
Discipline Specific Elective - (DSE) Credit:6

Course Objective

The course is focused on design of assembler and basic compiler. The course covers topics like absolute loader, relocating loader and dynamic linking.

Course Learning Outcomes

On successful completion of the course, the students will be able to:

1.

describe the working of assemblers and compilers.

2. use Lex/Yacc for building basic compiler.
3. develop a two pass Assemblers.
4. describe the role of the loaders, linkers and relocatable programs.

Unit 1

Assemblers & Loaders, Linkers: One pass and two pass assembler, design of an assembler, Absolute loader, relocation and linking concepts, relocating loader and Dynamic Linking.

Unit 2

Introduction: Overview of compilation, Phases of a compiler.

Unit 3

Lexical Analysis: Role of a Lexical analyzer, Specification and recognition of tokens, Symbol table, lexical Analyzer Generator.

Unit 4

Parsing & Intermediate representations: Bottom up parsing- LR parser, yacc,three address code generation, syntax directed translation, translation of types, control statements

Unit 5

Storage organization & Code generation: Activation records, stack allocation, Object code generation

Practical

Practicals based on topics covered in theory.

References

1. Santanu Chattopadhyaya, System Software,PHI,2011.
2. AlfredV.Aho, MonicaS.Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools, 2ndedition, Prentice Hall.

Additional references:

1. D. M. Dhamdhere,Systems Programming, TataMcGrawHill, 2011(reprint ,2015).
2. Leland Beck,D.Manjula, System Software:An Introduction to System Programming,3rd edition, Pearson Education.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

compilers, Lexical analyzer, Syntax directed translation,assembler,loader,linker.

Unix Network Programming
(BHCS18C)
Discipline Specific Elective - (DSE) Credit:6

=""">

Course Objective

This course introduce the concepts of Internet protocols, ports used during communication, Client/Server concepts and various transport protocols used in computer network applications and services. The objectives is to equip the students with technical knowlege of it comprises of the study of the sockets used with TCP and UDP include IPV4 & IPV6. The topic also covers basic name and address conversions. Sockets include elementary, advanced, routed and raw. Topics related to I/O multiplexing include non-blocking and signal-driven approaches. Designing of Client/Server programs includes Day time and Echo implementations. Both IPv4 and IPv6 protocols are also covered in detail. Multithreaded servers are discussed in detail. Implementation

of server as a daemon process is included.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. describe and analyse the various Internet Transport layer protocols used in TCP/IP AND UDP.
2. comprehend the concepts and structures of both TCP based connection-oriented and UDP based connection-less client server applications.
3. write various real-life client-server applications using socket programming.
4. modify, maintain and extend the present internet client-server applications and write any new type of internet applications to suit the current needs of Internet users.

Unit 1

Introduction: Basics of Client Server applications, Example of day time client server, concurrent servers, protocols, sockets, port numbers.

Unit 2

Connection-oriented and Connection-less client server Applications: Elementary TCP sockets – Socket, connect, bind, listen, accept, fork and exec function, close function, Socket Address Structures, Byte Ordering and Manipulation Functions, TCP Client and Server for Echo, Signal Handling in case of crashing and rebooting of server, Shutdown process.
function

Unit 3

Socket Options: Getsockopt and stockpot functions, Socket states, Generic socket option

Unit 4

Connection-oriented and connection-less Sockets: TCP-oriented basic concurrent client server applications, UDP oriented Echo client and server application, Handling of errors like lost datagram, Lack of flow control with UDP, determining outgoing interface with UDP.

Unit 5

Elementary name and Address conversions: Domain Name System, socket functions like gethostbyname, gethostbyname2, gethostbyaddr function, uname function, gethostname function, getservbyname and getservbyport functions.

Unit 6

Advanced Sockets: Daemon Processes, Multithreaded server, Raw sockets.

Practical

The Practicals shall be based on concepts taught in the course.

References

1. R. W. Stevens, B. Fenner, A. M. Rudoff, Unix Network Programming: The Sockets Networking API, 3rd edition, Vol.1, PHI, 2010.

Additional Resources:

1. A.S. Tanenbaum; Computer Networks, 5thedition, Pearson, 2012.
2. B.A. Forouzan, Data Communications and Networking, 4thedition, Tata McGraw Hill, 2006.
3. R. W. Stevens, Unix Network Programming, 1stedition, PHI, 2009.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Connection oriented sockets, connection less sockets, advanced sockets

---

Android Programming
(BHCS20A)
Skill-Enhancement Elective Course - (SEC) Credit:4

Course Objective

The paper provides an introduction to development of mobile application on android platform. The topics include the Android development environment, activities, fragments, user interfaces, intents, broadcast sender/receivers, services, notifications, SQLite database handling.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. describe characteristics of Android operating system
2. describe components of an android applications
3. design user interfaces using various widgets, dialog boxes, menus
4. define interaction among various activities/applications using intents, broadcasting, services
5. develop Android applications that require database handling

Unit 1

Introduction: Review to JAVA & OOPS Concepts, History of Android, Introduction to Android Operating Systems, Android Development Tools, Android Architecture, Android components including activities, view and view group, services, content providers, broadcast receivers, intents, parcels, instance state.

Unit 2

User Interface Architecture: application context, intents: explicit intents, returning results from activities, implicit intents, intent filter and intent resolution, and applications of implicit intents, activity life cycle, activity stack, application’s priority and its process’ states, fragments and its life cycle.

Unit 3

User Interface Design: Layouts, optimizing layout hierarchies, form widgets, text fields, button control, toggle buttons, spinners, images, menu, dialog.

Unit 4

Broadcast receivers, notifications and services: Broadcast sender, receiver, broadcasting events with intents, listening for broadcasts with broadcast receivers, broadcasting ordered intents, broadcasting sticky intents, pending intents, creating notifications, setting and customizing the

notification tray UI. Create, start, and stop services, binding services to activities, using asynctasks to manage background processing, handler, looper and runnable

Unit 5

Database and Content provider: SQLite, Content Values and Cursors, creating SQLite databases, querying a database, adding, updating, and removing rows, Creating Content Providers, implement content provider’s queries and its usage.

References

1. Dawn Griffiths and David Griffiths, Head First Android Development, O'reilly, 2015.
2. Reto Meier, Professional Android™ 4 Application Development, John Wiley & Sons, Inc., 2012.

Additional Resources:

1. James C. Sheusi, Android Application Development for Java Programmers. Cengage Learning, 2013.
2. Bill Phillips, Chris Stewart, Brian Hardy and Kristin Marsicano, Android Programming: The Big Nerd Ranch Guide, Big Nerd Ranch, LLC., 2015.
3. Mark L. Murphy, The Busy Coder's Guide to Android Development, CommonsWare, 2018.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Android App Development, Activities, Fragments, User interfaces, Intents, Broadcast

sender/receivers, Services, Notifications, SQLite Database

Introduction to R Programming
(BHCS20B)
Skill-Enhancement Elective Course - (SEC) Credit:4

Course Objective

This course introduces statistical programming language R. The course covers data reading and its manipulation using R data structures. The course also covers different control structures and design of user-defined functions. Loading, installing and building package is covered.

Course Learning Outcomes

On successful completion of the course, students will be able to do following:

1. develop an R script and execute it
2. install, load and deploy the required packages, and build new packages for sharing and reusability
3. extract data from different sources using api and use it for data analysis
4. visualize and summarize the data
5. design application with database connectivity for data analysis

Unit 1

Introduction: R interpreter, Introduction to major R data structures like vectors, matrices, arrays, list and data frames, Control Structures, vectorized if and multiple selection, functions.

Unit 2

Installing, loading and using packages:

- To Read/write data from/in files, extracting data from web-sites

- To clean Data to remove NULL values, outliers and noise
- Transform data by sorting, adding/removing new/existing columns, centring, scaling and normalizing the data values, converting types of values, using string in-built functions
- Statistical analysis of data for summarizing and understanding data
- Visualizing data using scatter plot, line plot, bar chart, histogram and box plot

Unit 3

Designing GUI : Building interactive application and connecting it with database.

Unit 4

Building Packages.

Unit 5

Unit 6

Practical

Programs based on concepts covered in theory

References

1. Richard Cotton, Learning R: a step by step function guide to data analysis, O'reilly (SPD), first edition

Additional Resources:

2. Mark Gardener, Begining R: The statistical programming language, WILEY, 2017

3. Michael Lawrence, John Verzani, Programming Graphical User Interfaces in R, CRC press, 2016 (ebook)

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
-

Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

R data structures, flow control, packages, functions

Java Programming
(BHCS19B)
Skill-Enhancement Elective Course - (SEC) Credit:4

Course Objective

This course adds to the basic programming language skills acquired by the student in earlier semesters. The students are exposed to the advanced features available in Java such as exception handling, file handling, interfaces, packages and GUI programming. The focus is to equip the students with adequate knowledge of these relatively new features of Object oriented languages that have become as standard as looping, selection, etc.

Course Learning Outcomes

On successful completion of the course the student will be

1. implement Exception Handling and File Handling.
2. implement multiple inheritance using Interfaces.
3. logically organize classes and interfaces using packages.
4. use AWT and Swing to design GUI applications.

Unit 1

Review of Object Oriented Programming and Java Fundamentals Structure of Java programs, Classes and Objects, Data types, Type Casting, Looping Constructs.

Unit 2

Interfaces Interface basics; Defining, implementing and extending interfaces; Implementing multiple inheritance using interfaces Packages Basics of packages, Creating and accessing packages, System packages, Creating user defined packages

Unit 3

Exception Handling Using the main keywords of exception handling: try, catch, throw, throws and finally; Nested try, Multiple catch statements, Creating user defined exceptions

Unit 4

File Handling Byte Stream, Character Stream, File I/O Basics, File Operations

Unit 5

AWT and Event Handling The AWT class hierarchy, Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Creating GUI applications using AWT.

Unit 6

Swing Introduction to Swing, Swing vs. AWT, Hierarchy for Swing components, Creating GUI applications using Swing.

Practical

Programs based on the concepts covered in theory.

References

1. Herbert Schildt, Java: The Complete Reference, 10th Edition, McGraw-Hill Education, 2018

Additional Resources:

1. Cay S. Horstmann, Core Java - Vol. I – Fundamentals, 10th Edition, Pearson, 2017
2. Herbert Schildt and Dale Skrien, Java Fundamentals - A Comprehensive Introduction, McGraw Hill Education (India) Private Limited, 2012.
3. E Balagurusamy, Programming with JAVA: A Primer, 5th Edition, McGraw Hill Education (India) Private Limited, 2014

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Objects and classes, interfaces, exceptional handling, file handling

---

Web Design and development
(BHCS19A)
Skill-Enhancement Elective Course - (SEC) Credit:4

Course Objective

This course will introduce students to the fundamental concepts of web development. This course will equip students with the ability to design and develop a dynamic website using technologies like HTML, CSS, JavaScript, PHP and MySQL on platform like WAMP/XAMP/LAMP.

Course Learning Outcomes

On successful completion of the course, students will be able :

1. design and develop a website
2. use Front end technologies like HTML, CSS and JavaScript
3. use backend technologies like PHP and MySQL
4. work on platforms like WAMP/XAMP/LAMP

Unit 1

Introduction to Static and Dynamic Websites (Website Designing and Anatomy of Webpage)

Unit 2

Introduction to HTML and CSS (Basic Tags, Lists, Handling Graphics, Tables, Linking, Frames, Forms), Introduction to DOM

Unit 4

Introduction to JavaScript (Basic Programming Techniques & Constructs, GET/POST Methods, Operators, Functions, DOM Event handling, Forms Validation, Cookies), Inter-page communication and form data handling using JavaScript

Unit 5

Introduction to PHP (Working, Difference with other technologies like JSP and ASP), PHP Programming Techniques (Data types, Operators, Arrays, Loops, Conditional statements, Functions, Regular expressions)

Unit 6

Form Data Handling with PHP, Database connectivity and handling using PHP-MySQL

References:

1. 1. Ivan Bayross, “Web enabled commercial application development using HTML, JavaScript, DHTML and PHP”, BPB Publication 4th Edition, 2013.

2. 2. Steven Holzner, "PHP: The Complete Reference Paperback", McGraw Hill Education (India), 2007.

3. 3. Timothy Boronczyk, Martin E. Psinas, "PHP and MYSQL (Create-Modify-Reuse)", Wiley India Private Limited, 2008.

4. 4. Robin Nixon, "Learning PHP, MySQL, JavaScript, CSS & HTML5", 3rd Edition Paperback, O'reilly, 2014.

Additional Resources

1. 1. Luke Welling, Laura Thompson, PHP and MySQL Web Development", 4th Edition, Addition Paperback, Addison-Wesley Professsional,2008.

2. 2. David Sklar, Adam Trachtenberg, "PHP Cookbook: Solutions & Examples for PHP Programmers", 2014.

Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations.

Keywords

Static and dynamic websites, form handling, database connectivity.