Computer Networks
(BSCS04)
Core Course - (CC) Credit:6

Course Objective

This course provides an overview of the concepts of data communication and computer networks. Network topologies and their characteristics, different type of networks, transmission media along with their limitations and use, different protocols used in application layer are covered.

Course Learning Outcomes

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

1. understand the basics of data communication.
2. differentiate between various types of computer networks and their topologies.
3. understand the difference between the OSI and TCP/IP protocol suit.
4. explain merits and demerits of different types of communication media.
5. distinguish between different types of network devices and their functions.
6. use IP addressing and understand the need of various application layer protocols.

Unit 1

Introduction: Introduction to data communications and networking, use of Computer Networks, classification of networks, OSI model, function of the layers, TCP/IP Protocol suite.

Unit 2

Network Topologies: Bus, star, ring, mesh, tree, hybrid topologies with their features, advantages and disadvantages of each type. Transmission Modes: simplex, half duplex and full duplex.

Unit 3

Transmission Media: Guided Media (Wired) (Twisted pair, Coaxial Cable, Fiber Optics. Unguided Media (Radio Waves, Infrared, Micro-wave, Satellite).

Unit 4

Data Communication and Switching Techniques: Framing, flow control, error control, circuit switching, message switching, packet switching, routing.

Unit 5

Switching Devices: Repeaters, hubs, switches, bridges, routers, gateways. Multiplexing: (FDM, WDM, TDM)

Unit 6

Internet: Internet Service Providers (ISP), internet addressing system: IP address with their classification and notation, application layer protocols: (DNS, URL, WWW, FTP, SMTP, HTTP, TELNET), web pages, introduction to HTML.

Practical

Practicals shall be based on concepts taught in the paper.

References

1. Douglas E. Comer, Computer Networks and Internet, 6th edition, Pearson Publication, 2015.

Additional Resources:

1. Behrouz A Forouzan, Data Communications and Networking, 5th edition, McGraw Hill, Indian Reprint 2017
2. Andrew S. Tannenbaum, David J. Wetherall, Computer Networks, Pearson Publication, 5th edition

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Data communication, Computer Networks, Switching, Internet Protocol, IP address

Computer System Architecture
(BSCS02)
Core Course - (CC) Credit:6

Course Objective

The course will introduce students to the fundamental concepts of digital computer organization, design and architecture. It aims to develop a basic understanding of the design of a computer system.

Course Learning Outcomes

Upon 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.

2. differentiate between combinational circuits and sequential circuits

3. represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.

4. determine various stages of instruction cycle, various instruction formats and instruction set.

5. describe interrupts and their handling.

6. explain how CPU communicates with memory and I/O devices.

Unit 1

Digital Logic Gates, Flipflops and their characterstic table, Logic circuit simplification using Boolean Algebra and Karnaugh Map, Don’t Care conditions.
Combinational Circuits, Sequential Circuits.

Unit 2

Digital Components: Decoders, Encoders, Multiplexers, Binary Adder, Binary Adder-Subtractor, Binary Incrementer, Registers and Memory Units

Unit 3

Binary representation of both numeric and alphanumeric data, representation of numeric data in different number systems (Binary, Octal, Decimal and Hexadecimal), conversion from one number system to another, complements, representation of decimal numbers, representation of signed and unsigned numbers, addition and subtraction of signed and unsigned numbers and overflow detection.

Unit 4

Arithmetic and logical micro-operations, instruction format, micro programmed control vs hardwired control, 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

Instruction format illustration using single accumulator organization, general register organization and stack organization, zero-address instructions, one-address instructions, two-address instructions and three-address instructions, Addressing Modes

Unit 6

Peripheral Devices, I/O interface, I/O vs. Memory Bus, Isolated I/O, Memory Mapped I/O, Direct Memory Access

References

1. M. Mano, Computer System Architecture, Third Edition, Pearson Education 1992

Additional Resources:

1. W. Stallings, Computer Organization and Architecture Designing for Performance, 8th Edition, Prentice Hall of India, 2009

2.M. Mano, Digital Design, Pearson Education Asia

3. Linda Null, Julia Lobur, The Essentials of Computer Organization and Architecture, Third Edition, Reprint 2013, Jones and Bartlett

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Combinational circuits, Data representation, Interrupts, I/O interface

Operating Systems
(BSCS03)
Core Course - (CC) Credit:6

Course Objective

This course introduces Operating System concepts and its importance in computer system. It focuses on the basic facilities provided in modern operating systems.

Course Learning Outcomes

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

1. understand the rationale behind the current design and implementation decisions in modern Operating Systems by considering the historic evolution.

2. identify modules of the operating systems and learn about important functions performed by operating system as resource manager.

3. use the OS in a more efficient manner.

Unit 1

Introduction : Operating systems, System software, Operating system resources, Operating System as resource manager.

Unit 2

Types of Operating Systems and Organization: Multiprogramming, batch, time sharing operating systems, personal computers & workstations. Basic OS functions, mechanisms of requesting operating system services – system calls and system programs.

Unit 3

Processor Management: Distinction between program and process, process address space, process states, process scheduling algorithms, process schedulers.

Unit 4

Memory Management: Mapping logical address space to physical address space, fixed partition, variable partition, paging, segmentation, virtual memory.

Unit 5

File and Input/Output Device Management: Classifications of I/O devices, I/O handling, file systems services, directory structure, disk storage.

Practical

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

References

1. Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.

Additional Resources:

1. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
2. G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.
3. W. Stallings, Operating Systems, Internals & Design Principles, 5th Edition, Prentice Hall of India, 4. 2008.
4. M. Milenkovic, Operating Systems - Concepts and design, Tata McGraw Hill 1992.

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Memory Management, Process Management, File Management, Virtual memory

Problem Solving using Computers
(BSCS01)
Core Course - (CC) Credit:6

Course Objective

This course is designed as the first course in programming to develop problem solving skills. The course focuses on modularity, reusability, code documentation, and debugging skills. It also introduces the concept of object-oriented programming.

Course Learning Outcomes

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

1. describe the components of a computer and the notion of an algorithm.
2. apply suitable programming constructs and data structures to solve a problem.
3. develop, document, and debug modular python programs.
4. use classes and objects in application programs.
5. use files for I/O operations.

Unit 1

Computer Fundamentals and Problem Solving: Basic Computer Organization: CPU, memory, I/O Units. Problem solving using computer, notion of an algorithm.

Unit 2

Introduction to Python Programming: Python interpreter, using python as calculator, python shell, indentation, identifiers and keywords, literals, strings, arithmetic, relational and logical operators.

Unit 3

Creating Python Programs: Input and output statements, defining functions, control statements default arguments, errors and exceptions.

Unit 4

Inbuilt Data Structures: strings, lists, sets, tuples, nested lists, built-in functions, dictionary and associated operation.

Unit 5

Object Oriented Programming: Introduction to Classes, Objects and Methods, Standard Libraries, File handling through libraries.

Unit 6

Sorting and Searching: Iterative and Recursive methods.

Practical

Programs based on concepts covered in theory.

References

1. Allen B. Downey. Think Python–How to think like a Computer Scientist. Green Tea Press, 2008.
2. Michael Urban, Joel Murach, Python Programming, Shroff, 2018

Additional Resources:

1. Y. Daniel Liang. Introduction to Programming using Python. Pearson, 2013.
2. Sheetal Taneja, Naveen Kumar. Python Programmine - A modular Approach. Pearson, 2018.
3. John V. Guttag. Itroduction to computation and programming using Python. MIT Press, 2013.

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Problem Solving, Control structure, Functions, Strings, Lists, Object Oriented Programming

Analysis of Algorithms
(BSCS06B)
Discipline Specific Elective - (DSE) Credit:6

Course Objective

The course provides techniques for design and analysis of algorithms. Topics include sorting, searching, heaps, divide and conquer, greedy and dynamic programming, and graph algorithms.

Course Learning Outcomes

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

1. understand the idea of algorithm analysis.
2. understand characteristics of searching and sorting algorithms and compare efficiency of different solutions for an application at hand.
3. model simple problems as graphs and solve those using graph algorithms.

Unit 1

Mathematical Preliminaries: Growth of functions, Asymptotic notation, standard notations and common functions. Recurrences: The substitution method for solving recurrences, recursion-tree method for solving recurrences, the master method for solving recurrences.

Unit 2

Divide and Conquer Algorithms: General method, binary search, merge sort, quicksort algorithms, space and running time analysis of the algorithms.

Unit 3

Sorting Algorithms: Insertion sort, Heapsort, Sorting in linear time.

Unit 4

Graph Algorithms: Representations of graphs, Breadth-first search, Depth-first search, topological sort.

Unit 5

Greedy Algorithms and dynamic programming: Minimum spanning tree, shortest path in a graph, 0/1 knapsack problem and fractional knapsack problem.

Unit 6

Practical

Programs based on the concepts covered in the theory.

References

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms,Third Edition, PHI, 2015.

Additional Resources:

1. Sara Basse, A.V. Gleder, Computer Algorithm – Introduction to Design and Analysis, Pearson, Third edition , 1999.

2. Jon Kleinberg, Eva Tardos, Algorithm Design. Pearson, 2013 (Paperback edition)

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Asymptotic Notation, Divide and Conquer Algorithms, Sorting Algorithms, Graph Algorithms, Greedy Algorithms, Dynamic Programming

Data Structures
(BSCS05A)
Discipline Specific Elective - (DSE) Credit:6

Course Objective

The course introduces the students to the fundamentals of data structures. Students will learn about arrays, stacks, queues, linked lists, recursion and trees.

Course Learning Outcomes

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

1. demonstrate a thorough understanding of the behaviour of basic data structures.
2. implement data structures efficiently in programming language C++.
3. demonstrate an understanding of recursion by applying recursive techniques to solve problems.

Unit 1

Arrays and Sorting: Single and multi-dimensional arrays, sparse matrices, different sorting methods including bubble, selection, insertion, merge, quick sort, linear and binary searching.

Unit 2

Stacks: Implementing stack using array, prefix, infix and postfix expressions, application of stacks for conversion of infix to prefix and postfix expressions, evaluation of postfix expressions.

Unit 3

Queue: Implementing simple queue, circular queues and priority queues using array.

Unit 4

Linked Lists: Single, double and circular lists, implementing stack and queue using linked lists.

Unit 5

Recursion: Recursive solutions to simple problems and their implementation, advantages and limitations of recursion.

Unit 6

Trees: Introduction to tree as a data structure, binary trees, binary search tree- creation and traversal techniques.

Practical

Programs based on the concepts covered in theory.

References

1. 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

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz

- End semester exam
- Internal assessment

Keywords

Arrays, linked lists, stacks, queues, tree, recursion

Database Management Systems
(BSCS05B)
Discipline Specific Elective - (DSE) Credit:6

Course Objective

The course introduces the students to the fundamentals of database management systems and methods to store and retrieve data. The course would give students hands-on practice of structured query language in a relational database management system.

Course Learning Outcomes

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

1. use database management system to manage data.
2. create entity relationship diagrams for modeling real-life situations and design the database schema.
3. use the concept of functional dependencies to remove data anomalies and arrive at normalized database design.
4. write queries using relational algebra and SQL.

Unit 1

Introduction to DBMS: Introduction to Database Management Systems, characteristics of database approach, data models, DBMS architecture and data independence.

Unit 2

Conceptual Modelling using ERD and EERD: Entity Relationship (ER) and Enhanced ER (EER) modeling, entity types, relationships, relationship constraints, and object modeling.

Unit 3

Relational Data Model and Relational Algebra: Relational data model concepts, relational constraints, queries in relational algebra.

Unit 4

Introduction to SQL: Data definition and data manipulation queries in SQL.

Unit 5

Database Design: Mapping of ER and EER diagrams to relational database, functional dependencies, Normalization and normal forms up to third normal form.

Practical

Practicals are based on Theory

References

1. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems (7th Ed.), Pearson Education. 2017.

Additional Resources:

1. R. Ramakrishanan, J. Gehrke, Database Management Systems (3rd Ed.), McGraw-Hill. 2002

2. A. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts (6th Ed.), McGraw Hill.

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

DBMS architecture, Data Independence, Entity modeling, Relational Data Model, SQL, Normalization

Digital Image Processing
(BSCS06A)
Discipline Specific Elective - (DSE) Credit:6

Course Objective

The course introduces the basic concepts and methodologies of digital image processing. The topics covered include image enhancement, spatial and frequency domain, image filtering, morphological image processing and image segmentation.
="">

Course Learning Outcomes

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

1. describe general terminology of Digital Image Processing and the roles of image processing systems in a variety of applications.
2. describe the basic issues and the scope (or principal applications) of image processing.
3. explain representation and manipulation of digital images, image acquisition, reading, writing, enhancement, displaying and segmentation and image Fourier transform.
4. examine various types of images, intensity transformations and spatial filtering.

Unit 1

Introduction: Fundamental steps in digital image processing (DIP), applications of DIP, components of image processing system, image types (binary, grayscale, color, truecolor, cartoon).

Unit 2

Digital Image Fundamentals : Elements of visual perception (Human eye, electromagnetic spectrum), Image acquisition, sampling and quantization, basic relationships between pixels.

Unit 3

Image Enhancement in spatial domain : Basic gray level transformations, histogram processing, smoothing and sharpening filters.

Unit 4

Image enhancement in frequency domain : DCT transform, enhancement filters in frequency domain, JPEG Image Compression.

Unit 5

Morphological Image processing: Erosion, dilation, opening , closing, Hit-or-miss transform, some basic morphological algorithms including boundary extraction, convex hull, thinning and thickening.

Unit 6

Image Segmentation : Detection of discontinuities, edge linking and basic thresholding.

Practical

Practicals will be based on the concepts covered in the course using MATLAB.

References

1. R C Gonzalez, R E Woods, Digital Image Processing, 3rd Edition, Pearson Education

Additional Resources:

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

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Visual perception, Image Segmentation, Fourier transform, DCT transform

Project Work / Dissertation
(BSCS06C)
Discipline Specific Elective - (DSE) Credit:6

Advanced Java Programming
(BSCS09A)
Skill-Enhancement Elective Course - (SEC) Credit:4

Course Objective

This course builds over basic Java language skills acquired by the student in earlier semester. The students are exposed to the advanced features available in Java such as exception handling, file handling, interfaces, packages and GUI programming.

Course Learning Outcomes

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

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

Unit 3

Packages: Basics of packages, creating and accessing packages, system packages, creating user defined packages.

Unit 4

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 5

File Handling : Byte stream, character stream, file I/O basics, file read/write operations.

Unit 6

GUI Programming : AWT classes, event handling.

Practical

Programs based on the concepts covered in theory.

References

1. Cay S. Horstmann, Core Java - Vol. I – Fundamentals, 10th Edition, Pearson, 2017.

Additional Resources

1. Herbert Schildt, Java: The Complete Reference, 10th Edition, McGraw-Hill Education, 2018.
2. E Balagurusamy, Programming with JAVA: A Primer, 5th Edition, McGraw Hill Education (India) Private Limited, 2014.

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Interface, Byte stream,Exception Handling, AWT, Event handling,

Android Programming
(BSCS10A)
Skill-Enhancement Elective Course - (SEC) Credit:4

Course Objective

The course is designed for students to help them learn how to develop android apps. They will also learn android architecture and key principles underlying its design.

Course Learning Outcomes

Learning outcome

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

1. describe the design of Android operating system.
2. describe various components of Android applications.
3. design user interfaces using various widgets, dialog boxes, menus.
4. design application with interaction among various activities/applications using intents.
5. develop application(s) with database handling.

Unit 1

Introduction: Review to JAVA & OOPS Concepts, introduction to Android operating systems and its development tools, android architecture along with components including activities, view and view group, services, content providers, broadcast receivers, intents, parcels, instance state. Android virtual device manager, Android SDK manager, Android emulator, Dalvik debug monitor service and debug bridge.

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, auto complete textview, edittext, images, image buttons, menu, dialog.

Unit 4

Broadcast receivers: Broadcast sender, receiver, broadcasting events with intents, listening for broadcasts with broadcast receivers, broadcasting ordered intents, broadcasting sticky intents, pending intents.

Unit 5

Database using SQLite: SQLite, content values and cursors, creating SQLite databases, querying a database, adding, updating, and removing rows.

References

Text Books:

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.

References Books:

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

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Android architecture, Android virtual device manager, Android SDK manager, Android emulator, Broadcast senders & receivers

C++ Programming
(BSCS08A)
Skill-Enhancement Elective Course - (SEC) Credit:4

Course Objective

The course introduces Object Oriented Programming Language C++ with the objective to use object oriented features to develop efficient programs. The focus is to equip the students with adequate high-level object-oriented programming features using C++.

Course Learning Outcomes

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

1. solve simple programming problems using iteration and selection, and basic constructs: structures, arrays and functions.
2. create classes and their objects and use access specifiers for data hiding depicting advantage of Abstraction.
3. construct classes for code reusability depicting advantage of Inheritance.

4. implement Function Overloading depicting advantage of Polymorphism.
5. create file, read/write from/to files.

Unit 1

Introduction to C++: Need and characteristics of Object-Oriented Programming, Structure of a C++ Program (main () function, header files, output, input, comments), compile and execute a simple program.

Unit 2

Data types and Expression: Keywords, built in data types, variables and constants, naming convention, Input-Output statements, expressions and operators, precedence of operators, typecasting, library functions.

Unit 3

Control Constructs in C++ : Decision making using selection constructs, looping constructs , control constructs.

Unit 4

User defined Data types and functions: User defined data types, defining and initializing structures, derived data types, defining and initializing single and multi dimensional arrays, and user defined functions, passing arguments to functions, returning values from functions, inline functions, default arguments.

Unit 5

Classes and Objects: Need of abstraction, encapsulation, inheritance and polymorphism, creating classes, objects as function arguments, modifiers and access control, constructors and destructors, Implementation of single level inheritance, implementation of polymorphism, function overloading.

Unit 6

File Handling: File I/O Basics, read and write operations.

Practical

Programs based on the concepts covered in theory.

References

1. Robert Lafore, Object Oriented Programming in C++, 4th Edition, SAMS Publishing

Additional Resources:

1. E Balaguruswamy, Object Oriented Programming with C++, 7th edition, McGraw-Hill Education, 2017
2. Yashavant P. Kanetkar, Let us C++, 2nd Edition, BPB Publishers, 2015

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

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Abstraction, Encapsulation, Inheritance and Polymorphism

Data Analysis using Python Programming
(BSCS07A)
Skill-Enhancement Elective Course - (SEC) Credit:4

Course Objective

The course enables students to analyse data using python. They will learn how to prepare data for analysis and create meaningful data visualisations. They will learn to use Pandas, Numpy and Scipy libraries to work with different data sets.

Course Learning Outcomes

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

1. develop a python script for data analysis and execute it.
2. install, load and deploy the required packages.
3. clean and prepare the data for accurate analysis.
4. analyse the data stored in files in different formats.
5. experiment with data visualization methods.

Unit 1

Introduction to Pandas, NumPy, SciPy: Introduction to Pandas DataFrames, Numpy multi-dimentional arrays, and SciPy libraries to work with different datasets.

Unit 2

Import and Export of Data: Installing, loading and using packages for importing and exporting data in Python.

Unit 3

Data Preprocessing and Transformation: Handling of missing data, Data cleaning and transformation.

Unit 4

Data Exploration: Exploring data using statistical methods: mean, median, mode, quantiles. Building contingency table. Basics of grouping data and Correlation.

Unit 5

Data Visualization: Scatter Plot, line graph, histogram, boxplot, line plots regression, word clouds, exporting plots as images.

Practical

Programs based on concepts covered in theory

References

1. Wes Mckinney, Python for Data Analysis, O’reilly (SPD), Second edition, 2017.

Additional Resources:

1. Joel Grus, Data Science from scratch, O’reilly (SPD), First edition, 2016
2. Jake VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, O’reilly (SPD), Second edition, 2016

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Data import and export, Cleaning, Transformation, Data Exploration, Data Visualization

Introduction to R Programming
(BSCS07B)
Skill-Enhancement Elective Course - (SEC) Credit:4

Course Objective

This course introduces statistical programming language R for data analysis. The objective is to expose the students to the strengths and capabilities of R for data analysis. It also encourages students to use open source softwares.

Course Learning Outcomes

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

1. develop an R script for data analysis and execute it.
2. install, load and deploy the required packages.
3. analyse the data stored in files in different formats.
4. identify suitable data visualization and exploration methods to answer a business question.
5. interpret the results of analysis.

Unit 1

Introduction to Programming Structures: R interpreter, Introduction to major R data structures like vectors, matrices, arrays, list and data frames, Flow control and loops, looping over list and array, user-defined functions.

Unit 2

File Handling: Installing, loading and using packages for reading data from files

Unit 3

Data Preprocessing and Transformation: Handling of missing data, Data cleaning and transformation

Unit 4

Data Exploration: Exploring data using statistical methods: mean, median, mode, quantiles. Building contingency table, correlation, co-variance.

Unit 5

Plotting Data: Data visualization using Scatter plot, Line graph, Histogram, Boxplot.

Practical

Practicals based on the concepts taught in theory class.

References

1. Richard Cotton, Learning R, A step by step function guide to data analysis, O'reilly (SPD), Sixth edition reprint, 2017.

Additional Resources:

1. Mark Gardener, Beginning R, The statistical programming language, WILEY, 2017

2. Paul Teetor, R Cookbook, O'reilly (SPD), Tenth edition reprint, 2017.

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Data exploration, Data analytics, Data visualization, Statistical analysis

Java Programming
(BSCS08B)
Skill-Enhancement Elective Course - (SEC) Credit:4

Course Objective

This course introduces fundamental concepts of Object Oriented Programming using Java. Basic concepts such as data types, expressions, control structures, functions and arrays are covered. Students are exposed to extensive Java programming to solve practical programming problems.

Course Learning Outcomes

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

1. develop and execute Java programs using iteration and selection.
2. create classes and their objects.
3. implement OOPS concepts to solve problems using JAVA

Unit 1

Introduction to Java: Features of Java, JDK environment, structure of Java programs

Unit 2

Programming Fundamentals: Data types, variables, operators, expressions, arrays, keywords, naming convention, decision making constructs , iteration, type casting, methods.

Unit 3

Object Oriented Programming Overview: Abstraction, encapsulation, inheritance, polymorphism.

Unit 4

Classes and Objects: Creating classes and objects, modifiers and access control, constructors, implementation of single and multilevel inheritance, implementation of polymorphism using overloading, overriding and dynamic method dispatch.

Unit 5

Strings: String class methods, string buffer methods.

Practical

Programs based on the concepts covered in theory.

References

Cay S. Horstmann, Core Java - Vol. I – Fundamentals, 10th Edition, Pearson, 2017.

Additional Resources:

1. Herbert Schildt, Java: The Complete Reference, 10th Edition, McGraw-Hill Education, 2018.
2. Herbert Schildt and Dale Skrien, Java Fundamentals - A comprehensive Introduction, TMH.
3. E Balagurusamy, Programming with JAVA: A Primer, 5th Edition, McGraw Hill Education (India) Private Limited, 2014.

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Abstraction, encapsulation, inheritance, polymorphism.

PHP Programming
(BSCS10B)
Skill-Enhancement Elective Course - (SEC) Credit:4

Course Objective

This course will introduce server side scripting to students through PHP programming language. They will learn to design web applications with a specific functionality, and dynamic websites requiring handling/processing data input by users.

Course Learning Outcomes

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

1. use different data types and control structures in PHP.
2. handle arrays and strings in PHP.
3. create dynamic interactive web pages with PHP.

4. use PHP built-in functions as well as define custom functions.
5. perform data validation in PHP.
6. manipulate and manage a database using PHP.

Unit 1

Introduction: Introduction to three tier web application development, front end, business layer and back end connectivity, role of PHP in web application development, software requirements.

Unit 2

Starting PHP Programming: Basics of PHP programming, variables, scope of a variable, expressions, operators, operator precedence, simple procedural scripts, decision making based on conditions, case structure, loops.

Unit 3

Modular Programming: Functions and objects, Passing parameters.

Unit 4

Strings and Arrays: Creating and accessing strings, built-in functions for string and string formatting, creating index based and associative array, accessing array elements.

Unit 5

Forms and form processing: Capturing form data, GET and POST form methods, processing of form data, and use of regular expressions.

Unit 6

Integrating PHP & DBMS: Connecting PHP and DBMS, creating database, defining database structure and accessing data stored in tables using PHP.

Practical

Programs based on the concepts covered in theory.

References

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

Additional Resources:

1. Luke Welling, Laura Thompson, PHP and MySQL Web Development, 4th Edition, Addition

Paperback, Addison-Wesley Professional, 2008.
2. David Sklar, Adam Trachtenberg, PHP Cookbook: Solutions & Examples for PHP Programmers, 2014.
3. Steven Holzner, PHP: The Complete Reference Paperback, McGraw Hill Education (India), 2007.
4. Timothy Boronczyk, Martin E. Psinas, PHP and MYSQL (Create-Modify-Reuse), Wiley India Private Limited, 2008.

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Server-side scripting, Web applications, Dynamic Websites, Database integration

Web Design using HTML5
(BSCS09B)
Skill-Enhancement Elective Course - (SEC) Credit:4

Course Objective

The course introduces the basics of HTML5 including CSS styling. It helps students learn how to plan and design effective web pages and producing effective websites.

Course Learning Outcomes

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

1. define the principles and basics of Web page design.
2. recognize the elements of HTML.
3. apply basic concepts of CSS.
4. publish the web pages.

Unit 1

Introduction: Introduction to HTML: What is HTML, HTML Documents, Basic structure of an HTML document, creating an HTML document, markup tags, heading-paragraphs, line breaks, HTML tags.

Unit 2

Elements of HTML: Introduction to elements of HTML, Working with Text, Working with Lists, Tables and Frames, Working with Hyperlinks, Images and Multimedia, Working with Forms and controls.

Unit 3

Introduction to Cascading Style Sheets: Concept of CSS, Creating Style Sheet, CSS Properties, CSS Styling(Background, Text Format, Controlling Fonts), Working with block elements and objects, Working with Lists and Tables, CSS Id and Class, Box Model(Introduction, Border properties, Padding Properties, Margin properties).

Unit 4

CSS Advanced: CSS Advanced(Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute sector), CSS Color.

Unit 5

Web Designs: Creating page Layout and Site Designs.

Unit 6

Introduction to Web Publishing or Hosting: Creating the Web Site, Saving the web site, Working on the web site, Creating web site structure, Creating Titles for web pages, Themes-Publishing web sites.

Practical

Practical exercises based on concepts mentioned in theory using relevant software.

References

1. Anne Boehm and Zak Ruvalcaba, Munarch’s HTML5 and CCS3, 4th Edition, 2018.

Additional Resources:

1. Jessica Minnick, Web Design with HTML5 and CSS3, 8th Edition, Cengage Learning, 2015.

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Assessment Methods

- Assignments, presentations, viva, quiz
- End semester exam
- Internal assessment

Keywords

Web Design, HTML, CSS, Web Publishing