

**GE-1**

**Information Technologies – 1**

**BS501**

**Theory**

2 Hours/Week

2 credits

**Unit – I**

Introduction to Computers: What is a Computer? Characteristics of Computers, Generations of Computers, Classification of Computers, Basic Computer Organization, Applications of Computers.

Input and Output Devices: Input Devices, Output Devices, Soft Copy Devices, Hard Copy Devices.

Computer Memory and Processors: introduction, Memory Hierarchy, Processor Registers, Cache Memory, Primary Memory, Secondary Storage Devices, Hard Disks, Optical Drives, USB Flash Drives, Memory Cards.

**Unit – II**

Computer Software: Introduction, Classification of Computer Software, System Software, Applications Software, Firmware, Middleware, Acquiring Computer Software.

Operating Systems: Introduction, Evolution of OS, Process Management, Memory Management, File Management, Device Management, Security Management, Command Interpreter, Windows, Linux.

**Text** Reema Thareja, *Fundamentals of Computers*

**References** P. K. sinha, *Computer Fundamentals*  
Anita Goel, *Computer Fundamentals*  
V. Rajaraman, *Fundamentals of Computers*  
E. Balagurusamy, *Fundamentals of Computers*  
J. Glenn Brookshear, Dennis Brylow, *Computer Science An Overview*

**Note:** Student friendly video lectures pertaining to this course are available at <http://spoken-tutorial.org/>  
Teachers are advised to teach this courses in the computer lab itself, so that the interested students may derive some time to perform few programs their own.

SEC-3

[A]

Python – 1

BS502

Theory

2 Hours/Week

2 credits

**Unit – I**

Introduction to Python: Python, Features of Python, Execution of a Python Program, Viewing the Byte Code, Flavors of Python, Python Virtual Machine, Frozen Binaries, Memory Management in Python, Garbage Collection in Python, Comparisons between C and Python, Comparisons between Java and Python.

Writing Our First Python Program: Installing Python for Windows, Installing numpy, Setting the Path to Python, Writing Our First Python Program, Executing a Python Program, Getting Help in Python, Getting Python Documentation Help, Reopening the Python Program in IDLE.

Data types in Python: Comments in Python, Doc strings, How Python Sees Variables, Data types in Python, Built-in data types, bool Data type, Sequences in Python, Sets, Literals in Python, Determining the Data type of a Variable, What about Characters, User-defined Data types, Constants in Python, Identifiers and Reserved words, Naming Conventions in Python.

**Unit – II**

Operators in Python: Arithmetic Operators, Assignment Operators, Unary Minus Operator, Relational Operators, Logical Operators, Boolean Operators, Bitwise Operators, Membership Operators, Identity Operators, Operator Precedence and Associativity, Mathematical Functions.

Input and Output: Output statements, Input Statements, Command Line Arguments.

Control Statements: Control Statements, The if Statement, A Word on Indentation, The if ... else Statement, The if ... elif ... else Statement, The while Loop, The for Loop, Infinite Loops, Nested Loops, The else Suite, The break Statement, The continue Statement, The pass Statement, The assert Statement, The return Statement.

**Text**

R. Nageswara Rao, *Core Python Programming*, Dreamtech Press

**References**

Mark Lutz, *Learning Python*

Tony Gaddis, *Starting Out With Python*

Kenneth A. Lambert, *Fundamentals of Python*

James Payne, *Beginning Python using Python 2.6 and Python 3*

Paul Gries, *Practical Programming: An Introduction to Computer Science using Python 3*

**Note:**

Student friendly video lecturers pertaining to this course are available at

<http://spoken-tutorial.org/>

Teachers are advised to teach this courses in the computer lab itself, so that the interested students may derive some time to perform few programs their own.

SEC-3  
[B]**Computer Organization****BS502****Theory**

2 Hours/Week

2 credits

**Unit – I**

Latches and Flip-Flops: Introduction, Set-Reset Latch, Gated Latches, Edge-Triggered D Flip-Flop, S-R Flip-Flop, J-K Flip-Flop, T Flip-Flop, Flip-Flops with Additional Inputs, Asynchronous Sequential Circuits.  
 Registers and Counters: Registers and Register Transfers, Shift Registers, Design of Binary Counters, Counter Design Using D Flip-Flops, Counter Design Using S-R and J-K Flip-Flops, Derivation of Flip-Flop Input Equations.

**Unit – II**

Sequential Circuit Design: Design Example—Code Converter, Design of Iterative Circuits, Design of Sequential Circuits Using ROMs and PLAs, Simulation and Testing of Sequential Circuits, Computer-Aided Design.  
 Circuits for Arithmetic Operations: Serial Adder with Accumulator, Design of a Binary Multiplier, Design of a Binary Divider.

**Text**Charles H. Roth, Jr. and Larry L. Kinney, *Fundamentals of Logic Design (7e)***References**M. Morris Mano, Michael D. Ciletti, *Digital Design (4e)*A. Saha and N. Manna, *Digital Principles and Logic Design*M. Rafiquzzaman, *Fundamentals of Digital Logic and Microcontrollers (6e)*Elliott Mendelson, *Theory and Problems of Boolean Algebra and Switching Circuit*M. Morris Mano, Charles R. Kime, Tom Martin, *Logic and Computer Design Fundamentals*

**DSC–3E****Programming in Java****BS505**

<b>Theory</b>	3 Hours/Week	3 credits
<b>Practical</b>	2 Hours/Week	1 credit

**Unit – I**

Introduction: Java Essentials, JVM, Java Features, Creation and Execution of Programs, Data Types, Type Conversion, Casting, Conditional Statements, Loops, Branching Mechanism, Classes, Objects, Class Declaration, Creating Objects, Method Declaration and Invocation, Method Overloading, Constructors–Parameterized Constructors, Constructor Overloading, Cleaning-up unused Objects, Class Variables & Methods-static Keyword, this Keyword, One-Dimensional Arrays, Two-Dimensional Arrays, Command-Line Arguments, Inner Class.

Inheritance: Introduction, Types of Inheritance, extends Keyword, Examples, Method Overriding, super, final Keywords, Abstract classes, Interfaces, Abstract Classes Verses Interfaces.

**Unit – II**

Packages–Creating and Using Packages, Access Protection, Wrapper Classes, String Class, StringBuffer Class.

Exception: Introduction, Types, Exception Handling Techniques, User-Defined Exception.

Multithreading: Introduction, Main Thread, Creation of New Threads – By Inheriting the Thread Class or Implementing the Runnable Interface, Thread Lifecycle, Thread Priority, Synchronization.

Input/Output: Introduction, java.io Package, File Class, FileInputStream Class, FileOutputStream Class, Scanner Class, BufferedInputStream Class, BufferedOutputStream Class, RandomAccessFile Class.

**Unit – III**

Applets: Introduction, Example, Life Cycle, Applet Class, Common Methods Used in Displaying the Output.

Event Handling: Introduction, Types of Events, Example. AWT: Introduction, Components, Containers, Button, Label, Checkbox, Radio Buttons, Container Class, Layouts. Swing: Introduction, Differences between Swing and AWT, JFrame, JApplet, JPanel, Components in Swings, Layout Managers, JTable, Dialog Box.

Database Handling Using JDBC: Introduction, Types of JDBC Drivers, Load the Driver, Establish Connection, Create Statement, Execute Query, Iterate Resultset, Scrollable Resultset, Developing a JDBS Application.

**Text** Sachin Malhotra, Saurabh Choudhary, *Programming in Java (2e)*

**References** Bruce Eckel, *Thinking in Java (4e)*  
 Herbert Schildt, *Java: The Complete Reference (9e)*  
 Y. Daniel Liang, *Introduction to Java Programming (10e)*  
 Paul Deitel, Harvey Deitel, *Java: How To Program (10e)*  
 Cay S. Horstmann, *Core Java Volume I – Fundamentals (10e)*  
 C. Thomas Wu, *An introduction to object-oriented programming with Java (5e)*  
 Tony Gaddis, *Starting Out with Java From Control Structures Through Objects (6e)*  
 Jeanne Boyarsky, Scott Selikoff, *OCA: Oracle Certified Associate Java SE 8 Programmer–I Study Guide*

## Java Lab

BS505

**Practical**

2 Hours/Week

1 credit

Write java programs to find the following

- 1 a) largest of given three numbers b) reverses the digits of a number  
c) given number is prime or not d) GCD of given two integers
- 2 Write java programs that implement the following  
a) default constructor b) parameterized constructor c) constructor overloading
- 3 a) Write a java program to find the smallest of given list integers using array and scanner class.  
b) Write a java program for multiplication of two matrices.
- 4 a) Write a java program for demonstrating an inner classes or nested classes.  
b) Write a java program to implement method overloading, method overriding, dynamic method dispatch
- 5 Write a java program to implement single, multilevel, hierarchal, multiple, hybrid inheritances.
- 6 Write java programs that demonstrate the use of abstract, this, super, static, final keywords
- 7 a) Write a java program for creating a package and using a package.  
b) Write a java program to demonstrate the use of wrapper classes.
- 8 a) Write a java program using all five keywords of exception handling mechanism.  
b) Write a java program for creating customized (user) exception
- 9 a) Write a java program that checks whether a given string is a palindrome or not.  
b) Write a java program for sorting a given list of names in ascending order.
- 10 a) Write a java program to create a file, write the data and display the data.  
b) Write a java program that reads a file name from user and displays its information.
- 11 a) Write a java program for controlling main thread.  
b) Write a java program for creating new thread by extending Thread class.
- 12 a) Write a java program for creating new thread by implementing Runnable interface.  
b) Write a java program for thread synchronization.
- 13 a) Write a java program to create following AWT components: Button, Checkbox, Choice, and List.  
b) Write java programs to create AWT application using containers and layouts.
- 14 a) Write java programs to create a simple Applet and create swing based Applet.  
b) Write a java program to handle different types of events in a swing application.
- 15 Write java programs to create a swing application using swing components and layouts.
- 16 Write a java program to store and retrieve data from database using JDBC.

**Note :** Write the program using simple text editors (not IDE), compile and run from command prompt.

Encourage students to develop small java applications using IDE, like giving as assignment.

Write a small java application using some features of java.

**DSE–1E****Operating Systems****BS506**

<b>Theory</b>	3 Hours/Week	3 credits
<b>Practical</b>	2 Hours/Week	1 credit

**Unit – I**

Introduction: Computer-System Architecture, Computing Environments.

Operating-System Structures: Operating-System Services, User Interface for Operating-System, System Calls, Types of System Calls, Operating System Structure.

Process Management: Process Concept, Process Scheduling, Operations on Processes, Inter process Communication, Examples–Producer-Consumer Problem.

Process Synchronization: Critical-Section Problem, Peterson’s Solution, Synchronization, Semaphores, Monitors.

**Unit – II**

CPU Scheduling: Concepts, Scheduling Criteria, Scheduling Algorithms.

Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

**Unit – III**

Main Memory: Introduction, Swapping, Contiguous Memory Allocation, Segmentation, Paging.

Virtual Memory: Introduction, Demand Paging, Page Replacement, Allocation of Frames, Thrashing.

Mass-Storage Structure: Overview, Disk Scheduling, RAID Structure.

File Systems: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, Protection.

File System Implementation, Directory Implementation, Allocation Methods, Free-Space Management.

**Text** Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, *Operating System Concepts (9e)*

**References** Naresh Chauhan, *Principles of Operating Systems*  
 Thomas W. Doeppner, *Operating Systems in Depth*  
 Andrew S. Tanenbaum, *Modern Operating Systems*  
 William Stallings, *Operating Systems – Internals and Design Principles*  
 Dhananjay M. Dhandhere, *Operating Systems – A Concept Based Approach*

## Operating Systems Lab

BS506

**Practical**

2 Hours/Week

1 credit

- a) Use vi editor to create different files, writing data into files, modifying data in files.
- 1 b) Use different types of Unix commands on the files created in first program.
- 2 Write shell programs using 'case', 'then' and 'if' & 'else' statements.
- 3 Write shell programs using while, do-while and for loop statements.
  - a) Write a shell script that accepts two integers as its arguments and computes the value of first number raised to the power of the second number.
- 4 b) Write a shell script that takes a command -line argument and reports on whether it is directory, a file, or something else.
  - a) Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers..
- 5 b) Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
  - a) Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
- 6 b) Develop an interactive script that ask for a word and a file name and then tells how many times that word occurred in the file.
- 7 Write a program that simulate the following Unix commands like ls, mv, cp.
- 8 Write a program to convert upper case to lower case letters of a given ASCII file.
- 9 Write a program to program to search the given pattern in a file.
- 10 Write a program to demonstrate FCFS process schedules on the given data.
- 11 Write a program to demonstrate SJF process schedules on the given data.
- 12 Write a program to demonstrate Priority Scheduling on the given burst time and arrival times.
- 13 Write a program to demonstrate Round Robin Scheduling on the given burst time and arrival times.
- 14 Write a program to implementing Producer and Consumer problem using Semaphores.
- 15 Write a program to simulate FIFO, LRU, LFU Page replacement algorithms.
- 16 Write a program to simulate Sequential, Indexed, and Linked file allocation strategies.

**Note**  
: Recommended to use Open Source Software like Fedora, Ubuntu, CentOS, etc...

**DSE–2E****Software Engineering****BS506**

<b>Theory</b>	3 Hours/Week	3 credits
<b>Practical</b>	2 Hours/Week	1 credit

**Unit – I**

Software Engineering – Introduction, Program Versus Software, Software Engineering, Software Development Process and its Stages, Generic Software Development Process Models, Code of Ethics and Professional Practice, Software Development and Maintenance Cost Breakup.

Requirement Engineering Processes – Requirement Engineering Process, Feasibility Study, Cost and Benefit Analysis, Requirement Specification, Characteristics of a Good Requirement and Validation Techniques, Requirements Management Planning, Process of Requirement Change Management.

Software Requirement Specifications – Introduction, Stakeholder Analysis, Software Requirements Document, IEEE Standard of Software Requirement Specifications, Organizing Functional Requirements, Traceability and Validation of Specifications.

**Unit – II**

Architectural Styles – Introduction, Architecture Styles, Object Oriented Architecture, Inter-organizational Communication, Cloud Computing Architecture Style, Core, Configurable and Customizable Architecture, Design Models, Architectural Design Principles.

Object Oriented System Analysis – Introduction, Object Oriented Design, Object Oriented Design Models, Object Oriented Analysis, Data Modeling, Comparison Between Top Down Structured and Object Oriented Analysis, Description of Logical and Static Modeling, Identification of Class Relationships.

Object Oriented Design Using UML – Introduction, Sequence Diagram, State Machine Diagram, Timing Diagram, Describing Detailed Object Oriented Design, Decision Tree and Decision Table, Composite Structure Diagram, Generating Test Cases, Moving Towards Physical Design, Structured Methods.

**Unit – III**

Software Development – Introduction, Good Coding Practices, Code Reuse, Design Pattern, Generator Based Reuse, Application/Software Developed on Product Lines Approach, Component Based Software Engineering, Agile Methods.

Verification, Validation and Software Testing – Introduction, Software Verification and Validation Process, Software Testing, System Testing, Object Oriented Testing Strategy, Test Cases, Equivalence Partitioning (Black Box Testing), Art of Debugging.

Measurement and Metrics for Assessing Software Quality – Introduction, ISO 9126 Quality Standards, Quality Management Models, Ways to Build Quality in Software, Software Quality Control and Metrics, Defect Density Metrics, Chidamber and Kemerer Metric Suites for Object Oriented System, Class Coupling Metric-Coupling Between Objects, Monitoring Dynamic Quality Attributes (Visible Externally) of a Software.

**Text** Rajesh Narang, *Software Engineering: Principles and Practices*

**References** Ian Sommerville, *Software Engineering*  
 R. Mall, *Fundamentals of Software Engineering*  
 Pankaj Jalote, *An Integrated Approach to Software Engineering*  
 Frank Tsui, Orlando Karam, Barbara Bernal, *Essentials of Software Engineering*  
 Roger S Pressman, B R Maxim, *Software Engineering – A Practitioner's Approach*  
 Grady Booch, *The Unified Modeling Language User Guide*



## Software Engineering Lab

BS506

**Practical**

2 Hours/Week

1 credit

### **Case Studies:**

- 1 Banking System
- 2 Hotel management system
- 3 Inventory Control System
- 4 Library management system
- 5 Railway Reservation System

**Choose any two of above case studies and do the following exercises for that case studies**

- 1 Write the complete problem statement
- 2 Write the software requirements specification document
- 3 Draw the entity relationship diagram
- 4 Draw the data flow diagrams
- 5 Draw use case diagrams
- 6 Draw activity diagrams for all use cases
- 7 Draw sequence diagrams for all use cases
- 8 Draw collaboration diagram
- 9 Assign objects in sequence diagrams to classes and make class diagram.

**Note :** To draw dataflow diagrams using Microsoft Visio Software, SmartDraw, etc...

To draw UML diagrams using Rational Rose Software, StarUML, etc...

The teacher should define the boundaries for the above case study problems and make the practice of problems mentioned.