

	Binary representation of data, representation of alpha data, representation of numeric data in different number systems, conversion between number systems, complements, representation of decimal numbers, representation of signed and unsigned numbers, addition and subtraction of signed and unsigned numbers and overflow detection.
7 - 11	Unit 4 - Operations and Control: Arithmetic and logical micro-operations, micro programmed control vs. hardwired control, instruction format, instruction set completeness, timing and control, instruction cycle, memory reference instructions and their implementation using arithmetic, logical, program control, transfer and input output micro operations, interrupt cycle.
12 - 13	Unit 5 - Instructions: Instruction format illustration using single accumulator organization, general register organization and stack organization, Addressing Modes, zero-address instructions, one-address instructions, two-address instructions and three-address instructions,
14 - 15	Unit 6 - Peripheral Devices: I/O interface, I/O vs. Memory Bus, Isolated I/O, Memory Mapped I/O, Direct Memory Access

Assessment Methods

Written tests, assignments, quizzes, presentations as announced by the instructor in the class.

Keywords

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

Programming in JAVA (BHCS03) Discipline Specific Core Course - (DSC)

Credit: 06

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.

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.

Detailed Syllabus

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

1. Design a class Complex having a real part (x) and an imaginary part (y). Provide methods to perform the following on complex numbers:
 1. Add two complex numbers.
 2. Multiply two complex numbers.
 3. toString() method to display complex numbers in the form: $x + i y$
2. Create a class TwoDim which contains private members as x and y coordinates in package P1. Define the default constructor, a parameterized constructor and override toString() method to display the co-ordinates. Now reuse this class and in package P2 create another class ThreeDim, adding a new dimension as z as its private member. Define the constructors for the subclass and override toString() method in the subclass also. Write appropriate methods to show dynamic method dispatch. The main() function should be in a package P.
3. Define an abstract class Shape in package P1. Inherit two more classes: Rectangle in package P2 and Circle in package P3. Write a program to ask the user for the type of shape and then using the concept of dynamic method dispatch, display the area of the appropriate subclass. Also write appropriate methods to read the data. The main() function should not be in any package.
4. Create an exception subclass UnderAge, which prints "Under Age" along with the age value when an object of UnderAge class is printed in the catch statement. Write a class exceptionDemo in which the method test() throws UnderAge exception if the variable age passed to it as argument is less than 18. Write main() method also to show working of the program.
5. Write a program to implement stack. Use exception handling to manage underflow and overflow conditions.
6. Write a program that copies content of one file to another. Pass the names of the files through command-line arguments.
7. Write a program to read a file and display only those lines that have the first two characters as '/' (Use try with resources).
8. Write a program to create an Applet. Create a frame as a child of applet. Implement mouseClicked(), mouseEntered() and mouseExited() events for applet. Frame is visible

when mouse enters applet window and hidden when mouse exits from the applet window.

9. Write a program to display a string in frame window with pink color as background.
10. Write a program to create an Applet that has two buttons named “Red” and “Blue”. When a button is pressed the background color of the applet is set to the color named by the button’s label.
11. Create an applet which responds to KEY_TYPED event and updates the status window with message (“Typed character is: X”). Use adapter class for other two events.
12. Create an applet with two buttons labeled ‘A’ and ‘B’. When button ‘A’ is pressed, it displays your personal information (Name, Course, Roll No, College) and when button ‘B’ is pressed, it displays your CGPA in previous semester.
13. Write a program that creates a Banner and then creates a thread to scrolls the message in the banner from left to right across the applet’s window.
14. Rewrite the applet programs using Swing.

References

1. Schildt, H. (2018). *Java: The Complete Reference*. 10th edition. McGraw-Hill Education.

Additional Resources:

1. Balaguruswamy E. (2014). *Programming with JAVA: A Primer*. 5th edition. India: McGraw Hill Education
2. Horstmann, C. S. (2017). *Core Java - Vol. I – Fundamentals* (Vol. 10). Pearson Education
3. Schildt, H., & Skrien, D. (2012). *Java Fundamentals - A Comprehensive Introduction*. India: McGraw Hill Education.

Course Teaching Learning Process

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

Tentative weekly teaching plan is as follows:

Week	Content
1	Review of Object Oriented Programming and Java Fundamentals Structure of Java programs, Classes and Objects, Data types, Type Casting, Looping Constructs
2	Interfaces Interface basics; Defining, implementing and extending interfaces; Implementing multiple inheritance using interfaces
3	Packages Basics of packages, Creating and accessing packages, System packages, Creating user defined packages
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
5	File Handling: Byte Stream, Character Stream, File I/O Basics, File Operations
6-9	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, Creating GUI applications using AWT
10-15	Swing: Introduction to Swing, Swing vs. AWT, Hierarchy for Swing components, Creating GUI applications using Swing, Creating GUI applications using Swing

Assessment Methods

Written tests, assignments, quizzes, presentations as announced by the instructor in the class.

Keywords

Objects and classes, interfaces, exceptional handling, file handling

Discrete Structures (BHCS04) Discipline Specific Core Course - (DSC)

Credit: 06

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: