

1	Multimedia: Introduction to multimedia, components, uses of multimedia, multimedia applications, virtual reality
2	Text: Fonts and faces, using text in multimedia, font editing and design tools, hypermedia and hypertext.
3	Images: Still images – bitmaps, vector drawing, 3D drawing and rendering.
4	Images (contd.): Natural light and colors, computerized colors, color palettes, image file formats.
5	Sound, Video and Animation: Digital Audio, MIDI audio, MIDI vs digital audio.
6	Sound, Video and Animation (contd.): Audio file formats, how video works, analog video, digital video.
7	Sound, Video and Animation (contd.): Video file formats, video shooting and editing.
8	Sound, Video and Animation (contd.): Principles of animation, animation file formats.
9	Sound, Video and Animation (contd.): Animation techniques.
10	Internet and Multimedia: WWW and HTML, Multimedia on the web.
11	Internet and Multimedia (contd.): Web servers, web browsers, web page makers and site builders.
12	Making Multimedia: Stages of a multimedia project, requirements to make good multimedia.
13	Making Multimedia (contd.): Hardware peripherals - connections, memory and storage devices.
14-15	Making Multimedia (contd.): Multimedia software and authoring tools.

Assessment Methods

- Unit-wise assignments, presentations, viva, quiz as announced by the instructor in the class.
- Internal assessment
- End semester exam

Keywords

Hypermedia, Hypertext, Bitmap, Video, Animation, HTML

**Programming in Java
(BACS05A)
Discipline Specific Elective - (DSE-I) Credit:6**

Course Objective

This course will introduce students to the fundamentals of computer programming in an object oriented framework using Java as programming language.

Course Learning Outcomes

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

1. understand the concepts of object-oriented software design.
2. develop, compile and run Java programs using basic programming constructs.
3. use object-oriented software design principles like inheritance and polymorphism .
4. use visibility modifiers (public, private, protected) to implement appropriate abstraction and encapsulation.

Unit 1

Introduction to Java programming: Java development environment, Java program structure.

Unit 2

Java Programming Constructs: Data types, variables, constants, scope and life time of variables, operators, expressions, type conversion and casting, control flow, conditional statements, loops, break and continue statements, arrays, command line arguments, methods.

Unit 3

Classes and Objects: Class, object, constructor, destructor, parameter passing, static fields and methods, access control, this reference, overloading of methods and constructors, garbage collection, accessibility modifiers.

Unit 4

OOPS concepts: Encapsulation, inheritance, polymorphism, dynamic binding, dynamic method dispatch, method overriding, final classes and methods, abstract classes and methods.

Practical

Practicals based on Java:

1. Write a java program to add, subtract, multiply and divide any two numbers entered by the user.
2. Write a java program to compute the square, cube of a number entered by a user using methods.
3. Write a java program to demonstrate the use of :
 - a. Bitwise operators.
 - b. Shift operators.
4. Write a java program to compute maximum of three numbers:
 - a. Using ternary operator.
 - b. Using if-else statement.
5. Write a menu driven program(using switch-case) which accepts a number as user input :
 - a. Checks whether the number is even or odd
 - b. Checks whether the number is prime
6. Write a menu driven program(using switch-case) which accepts a number as user input:
 - a. Prints sum of digits of the given number
 - b. Prints reverse of the given number
7. Write a program to display the first n terms of a Fibonacci series.
8. Write a method to compute the factorial of a number. Use this method to compute the sum of the series: $1 + x/1! + x^2/2! + x^3/3! + \dots + x^n/n!$

9. Write a java program that accepts a list of elements from the user in an array. Calculate the sum and average of the numbers entered. Accept the size of the array as command line argument.
10. Write a java program that accepts a list of elements from the user in an array and displays the elements in the ascending order.
11. Write a java program to create a class “Student” having:
 - a. Data members: RollNo, Name, Course
 - b. Methods: getData()-To retrieve values of data members
displayData()-To display values of data members
12. Write a java program to create a class “Room” having:
 - a. Data members: Length, Breadth, Height as dimensions of the room
 - b. Three constructors:
 - i. Having no parameter -that accepts values of data members from the user.
 - ii. Having one parameter –which specifies the same value of all dimensions.
 - iii. Having three parameters-which passes different values of all dimensions.
13. Write a java program to create a class “Student” having:
 - a. Private Data members: RollNo, Name, Marks1, Marks2, Marks3
 - b. Constructor: -To assign values to data members
 - c. Methods: -
 - i. CalculatePercentage()-To calculate percentage of marks
 - ii. CalculateDivision()- calculates division based on formula:
 - a) Ist Div if percentage is >60
 - b) IInd Div if percentage is between 50 and 60
 - c) IIIrd Div if percentage is <50
 - iii. displayMarksheet()-To display values of data members as well as percentage and division of student
14. Write a java program that calculates the area of circle, rectangle and triangle using method overloading
15. Create a class Employee containing information about employees of same organization. It should have following data members:
 - a. EmpId (Private),
 - b. CompanyName(static and final)
 - c. No-of-employees (static)

It should have a static method to calculate no. of employees in the organisation based on the number of objects of employees created and a display() method to display Number of employees in that organization.
16. Create a Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create and display objects of these two classes using Runtime Polymorphism.
17. Create a Circle class having data member radius. Create a subclass of circle called Cylinder having data members: radius and height. Both classes should have a method called area() that calculates their area.
18. Create an abstract class called shape having abstract method area() and two variables dimension1 and dimension2. Create two subclasses of shape, rectangle and triangle class which implement the method area().

References

1. Deitel, P., & Deitel, H. (2016). *Java How to Program* (10th Edition). Pearson.
2. Schildt, H. (2014). *Java The Complete reference* (9th Edition). Tata Mc Graw Hill.

Additional Resources

1. Balagurusamy, E. (2009). *Programming with Java: A primer* (4th Edition). Tata Mc Graw Hill.
 2. Horstmann, C. S. (2016). *Core Java Volume 1-Fundamentals* (10th Edition). Pearson Education.
-

Teaching Learning Process

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

Tentative weekly teaching plan is as follows:

Week	Topics
1	Introduction to Java programming: Java development environment, Java program structure.
2	Java Programming Constructs: Data types, variables, constants, scope and life time of variables.
3	Java Programming Constructs (contd.): Operators, expressions.
4	Java Programming Constructs (contd.): Type conversion and casting, control flow, conditional statements.
5	Java Programming Constructs (contd.): Looping constructs, break and continue statements.
6	Java Programming Constructs (contd.): Arrays, command line arguments, methods.
7	Classes and Objects: Classes and Objects.
8	Classes and Objects (contd.): Constructors and constructor, overloading.
9	Classes and Objects (contd.): Methods, parameter passing, static fields and methods, this reference.
10	Classes and Objects (contd.): Overloading methods, garbage collection.
11	Classes and Objects (contd.): Access control and accessibility modifiers.
12	OOP concepts: Inheritance, encapsulation, polymorphism.
13	OOP concepts (contd.): Dynamic binding, dynamic method Dispatch, method overriding.
14-15	OOP concepts (contd.): Final classes and methods, abstract classes and methods.

Assessment Methods

- Unit-wise assignments, presentations, viva, quiz as announced by the instructor in the class.
- Internal assessment
- End semester exam

Keywords

Object Oriented Programming, Overloading, Inheritance, Data abstraction, Encapsulation

Programming in Python (BACS05B) Discipline Specific Elective - (DSE-I) Credit:6

Course Objective

The course introduces programming in Python and develop Python based solutions for simple problems.

Course Learning Outcomes

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

1. select a suitable programming construct and inbuilt data structure for a situation.
2. develop and document modular python programs.
3. use classes and objects in application programs.

Unit 1

Introduction to Python: Structure of a Python program, Python interpreter/Python shell, identifiers and keywords, literals, strings, basic operators, input, output statements, Python standard libraries, notion of class, object.

Unit 2

Functions: Built in functions, function definition and calls, default parameter values.

Creating Python Programs: Input and output statements, control statements - branching, looping, exit function, break, continue, and pass, mutable and immutable structures, strings, lists, tuples and associated operations.

Unit 3

Control Structures: conditional statements, loops, exit, break and continue statements.

Unit 4

Classes: classes, objects and methods.