



Source: <https://www.datasciencecentral.com/wp-content/uploads/2021/10/8667507462.jpeg>

## 22CS203 OBJECT-ORIENTED PROGRAMMING THROUGH JAVA

Hours Per Week :

L	T	P	C
2	0	4	4

**PREREQUISITE KNOWLEDGE:** Classes, Objects and Inheritance.

### COURSE DESCRIPTION AND OBJECTIVES:

This course is about the fundamentals of Object-Oriented Programming (OOP) Concept and OOP-based software development methodology. Java as a class-based and pure OOP language is used to demonstrate and implement appropriate concepts and techniques. The students are exposed to the concepts, fundamental syntax, and the thought processes behind object-oriented programming. By end of the course, students will acquire the basic knowledge and skills necessary to implement object-oriented programming techniques in software development using Java.

### MODULE-1

#### UNIT-1

**6L+0T+12P=18 Hours**

##### INTRODUCTION:

History of Java, Byte code, JVM, Java buzzwords, OOP principles, Data types, Variables, Scope of variables, Operators, Control statements, Type conversion and casting, Arrays.

Concepts Of Classes and Objects: Introduction to methods, Method over loading, Constructors, Constructor overloading, Usage of static with data and method, Access control, this keyword, Garbage collection, String class, String Tokenizer.

#### UNIT-2

**10L+0T+20P=30 Hours**

##### INHERITANCE AND EXCEPTIONS:

**Inheritance:** Types of inheritance, Member access rules, Usage of super keyword, Method overriding, Usage of final keyword, Abstract classes, Interfaces - differences between abstract classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Packages-defining, creating and accessing a package, importing packages, access control in packages.

**Exception Handling:** Concepts of exception handling, Types of exceptions, Usage of try, catch, throw, throws and finally keywords, Built-in exceptions, User defined exception.

##### PRACTICES:

- There is a telecommunication company called "Powered Air" who have approached you to build their Interactive Voice Response (IVR) system. write a Java program and be able to provide the following menu (given below):

**Note:** User should provide an input for each menu display. Welcome to Powered Air service. What would you like to do?

- Know my balance.
- Know my validity date
- Know number of free calls available.
- More
- Prepaid Bill Request
- Customer Preferences
- GPRS activation
- Special Message Offers
- Special GPRS Offers
- 3G Activation
- Go back to Previous menu

You are free to display your own messages in this IVR.

- Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have read Attributes method to read length and width from user.

**Hint:** Area of rectangle = length \* width, Perimeter of rectangle = 2\*(length+width).

- Implement a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (use StringTokenizer class).
- Implement a java program to print all tokens of a string on the bases of multiple separators (use StringTokenizer class).
- Using inheritance, one class can acquire the properties of others. Consider a class Animal that has only one method "walk". Next, create a Bird class that also has a fly method. Finally, create a bird object that can both fly and walk.
- Using inheritance, Write the following code in your editor :
  1. A class named Arithmetic with a method named "add" that takes integers as parameters and returns an integer denoting their sum.
  2. A class named Adder that inherits from a superclass named Arithmetic.

**Note:** Your classes should not be Public.

- When a subclass inherits from a superclass, it also inherits its methods; however, it can also override the superclass methods (as well as declare and implement new ones). Consider the Sports class having methods getName()[which returns name of sport] and getNumberOfTeamMembers()[which returns noof team members] create a Soccer class that inherits from the Sports class. We can override the get Name method and return a different subclass-specific string and override getNumberOfTeamMembers method and return noof team members
- Implement a java program to create an abstract class named Shape that contains an empty method named number Of Sides ( ). Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method number Of Sides ( ) that shows the number of sides in the given geometrical figures.
- You are given an interface Advanced Arithmetic which contains a method signature int divisor\_sum(int n). You need to write a class called My Calculator which implements the interface. divisor\_sum function just takes an integer as input and return the sum of all its divisors. For example divisors of 6 are 1, 2, 3 and 6, so divisor\_sum should return 12. The value of n will be at most 1000.
- Implement a Java program for the following
  - a) Creation of simple package.
  - b) Accessing a package.
- Implement a Java program to read two numbers a,b from user and perform division a/b, if the user passes b value as zero, handle the exception using try and catch otherwise display the result.
- Create a class called Customer with data members account\_number, balance (initialize with 10000), and member functions print(), deposit(), and withdraw(). Print method display account number and balance. If withdraw amount is less than current balance while withdrawing, throw an exception "In Sufficient Funds". If the input is 1 do print. If the input is 2 withdraw (). If the input is 3 deposit. If the input is 4 terminate program.
- Implement a Java program which accepts age as input from the user and throws an exception "Not Eligible to Vote" when age is <=18 otherwise print "Eligible to Vote".

## MODULE-2

### UNIT-1

8L+0T+16P=24 Hours

#### MULTI THREDDING AND FRAMEWORK:

**Multithreading:** Concepts of multi threading, Differences between process and thread, Thread life cycle, Creating multiple threads using Thread class and Runnable interface, Synchronization, Thread priorities, Inter thread communication.

**Collection Framework:** Collections Overview, Collection Interfaces - List, Set, Map, List – Array List, Linked List, Vector, Set - HashSet, TreeSet, Map - HashTable, HashMap, accessing a collection via an Iterator, comparator, comparable.

#### SKILLS:

- ✓ To analyse and develop algorithm for real life problems using Java.
- ✓ Experience with developing and debugging programs in different IDEs.
- ✓ Develop multi-threaded applications.
- ✓ Creating web applications

**UNIT-2****8L+0T+16P=24 Hours****SWINGS:**

**GUI Programming With Swing:** Delegation event model-Events, Event sources, Event Listeners, Event classes, handling mouse and keyboard events.

**Exploring Swing Controls:** JLabel and Image Icon, JText Field, JButton, JCheckBox, JRadioButton, JTabbed Pane, JList, JCombo Box.

**PRACTICES:**

- Print in Order

Suppose we have a class:

```
public class Foo {
    public void first() { print("first"); }
    public void second() { print("second"); }
    public void third() { print("third"); }
}
```

The same instance of Foo will be passed to three different threads. Thread A will call first(), thread B will call second(), and thread C will call third(). Design a mechanism and modify the program to ensure that second() is executed after first(), and third() is executed after second().

**Note:** We do not know how the threads will be scheduled in the operating system, even though the numbers in the input seem to imply the ordering. The input format you see is mainly to ensure our tests' comprehensiveness.

**Example 1:**

Input: nums = [1,2,3]      Output: "firstsecondthird"

**Explanation:** There are three threads being fired asynchronously. The input [1,2,3] means thread A calls first(), thread B calls second(), and thread C calls third(). "firstsecondthird" is the correct output.

**Example 2:**

Input: nums = [1,3,2]      Output: "firstsecondthird"

**Explanation:** The input [1,3,2] means thread A calls first(), thread B calls third(), and thread C calls second(). "firstsecondthird" is the correct output.

- Flood Fill:

An image is represented by an m x n integer grid image where image[i][j] represents the pixel value of the image.

You are also given three integers sr, sc, and color. You should perform a flood fill on the image starting from the pixel image[sr][sc].

To perform a flood fill, consider the starting pixel, plus any pixels connected 4-directionally to the starting pixel of the same color as the starting pixel, plus any pixels connected 4-directionally to those pixels (also with the same color), and so on. Replace the color of all of the aforementioned pixels with color.

Return the modified image after performing the flood fill.

1	1	1		2	2	2
1	1	0	→	2	2	0
1	0	1		2	0	1

**Example 1:**

**Input:** image = [[1,1,1],[1,1,0],[1,0,1]], sr = 1, sc = 1, color = 2

**Output:** [[2,2,2],[2,2,0],[2,0,1]]

**Explanation:** From the centre of the image with position (sr, sc) = (1, 1) (i.e., the red pixel), all pixels connected by a path of the same color as the starting pixel (i.e., the blue pixels) are colored with the new color.

Note the bottom corner is not coloured 2, because it is not 4-directionally connected to the starting pixel.

**Example 2:**

**Input:** image = [[0,0,0],[0,0,0]], sr = 0, sc = 0, color = 0

**Output:** [[0,0,0],[0,0,0]]

**Explanation:** The starting pixel is already colored 0, so no changes are made to the image.

- **Count words in a given string**

The input parameter is a list of strings representing lines of text.

Count how often the word occurs in the text.

If the word “kitten” occurred in a text 23 times, then its entry would be “kitten - 23\n”. Return statistics as a String containing all the entries.

Omit all words which contain less than 4 letters and appear less than 10 (the words which are too small or too rare) The entries in the resulting String should be also sorted by their amount and then in alphabetical order if it is needed.

- Implement a Java program for handling mouse events when the mouse entered, exited, clicked, pressed, released, dragged and moved in the client area.
- Implement a Java program for handling key events when the key board is pressed, released, typed.
- Implement a Java swing program that reads two numbers from two separate text fields and display sum of two numbers in third text field when button “add” is pressed.
- Implement a Java program to design student registration form using Swing Controls. The form which having the following fields and button “save”. Form Fields are: Name, RNO, Mail id, Gender, Branch, and Address.
- Implement a java program using swings to design a multiple choice question having three options (use radio button) ,display the message using dialog box “Your answer is wrong” if the user selects wrong option otherwise display ,”Your answer is correct.”

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply object oriented concepts on real time scenarios.	Apply	1	1,2
2	Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes	Apply	1, 2	1,2
3	Design and develop Java applications to solve real world problems by using modern tools and collection framework	Create	2	3,5
4	Design and develop GUI based applications using swings for internet and system based applications.	Create	2	3,5

**TEXT BOOKS:**

1. Herbert Schildt, “Java the complete reference”, 12th Edition, McGraw Hill, Education, 2021.
2. M.T. Somashekara, D. S. Guru, K.S. Manjunatha, “Object-Oriented Programming with Java”, 1st Edition, PHI Learning, 2017.

**REFERENCE BOOKS:**

1. E. Balagurusamy, “Programming with Java”, 6th Edition, McGraw Hill, 2019.
2. Mark Lassooff, “ Java Programming for Beginners: Learn the fundamentals of programming with Java”, 1st Edition, Packt Publishing Limited,2017.
3. Philip Conrod, Lou Tylee, “ Learn Java GUI Applications : A JFC Swing Tutorial”, 11th Edition, Kidware Software, 2019.