



Object Oriented Programming using JAVA

(MCA292)

Assignment - 1

Basic Programming and Command Line Arguments

Write a java program for the following:

1. Write a Java Program to print your Name entered through the command line as an argument.
2. Write a Java program to convert Temperature from Fahrenheit to Celsius and vice versa.
3. Write a Java program to add two numbers.
4. Write a java Program to find the area and Perimeter of a rectangle.
5. Write a program in Java to find the maximum of three numbers.
6. Write a Java Program to check whether a given year is a leap year.
7. Create four different classes with three of them containing the function main.
Save the file with a different name than that of the class name and run each of the classes with the main function.
8. Write a java program to reverse a number entered as a command line argument.
9. Write a java Program to count the number of digits entered through the command line argument.
10. Write a java program to find all the multiples of 3 within a given range where the starting and ending value are entered through command line argument.



Object Oriented Programming using JAVA

(MCA292)

Assignment - 2

Constructors & Inheritance

1. Write a class, Grader, which has an instance variable, score, an appropriate constructor and appropriate methods. A method, letterGrade(), that returns the letter grade as O/E/A/B/C/F. Now write a demo class to test the Grader class by reading a score from the user, using it to create a Grader object after validating that the value is not negative and is not greater than 100. Finally, call the letterGrade() method to get and print the grade.
2. Write a class, Commission, which has an instance variable, sales; an appropriate constructor; and a method, commission() that returns the commission. Now write a demo class to test the Commission class by reading a sale from the user, using it to create a Commission object after validating that the value is not negative. Finally, call the commission() method to get and print the commission. If the sales are negative, your demo should print the message "Invalid Input".
3. For a Mobile Shop project, create "Telephone" class with details like mobile_id, model_name and available_quantity in "Phone" package. Inherit from this class and create a class for "smart_phone" with necessary details like enabled_5G, foldable and dual_screen in package "Smart". The customer executive tries to

display all smart_phone details (mobile_id, model_name, available_quantity, enabled_5G, foldable and dual_screen) and updates the quantity information, whenever the customer purchases the smart_phone. Write the necessary java programs to implement this scenario and test with user inputs.

4. An educational institution maintains a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown below. Write all the classes and define the methods to create the database and retrieve individual information as and when needed. Write a driver program to test the classes.

Staff (code, name)

Teacher (subject, publication) is a Staff

Officer (grade) is a Staff

Typist (speed) is a Staff

RegularTypist (remuneration) is a Typist

CasualTypist (daily wages) is a Typist.



Object Oriented Programming using JAVA

(MCA292)

Assignment - 3

Flow Control

1. The process of finding the largest value (i.e., the maximum of a group of values) is used frequently in computer applications. For example, a program that determines the winner of a sales contest would input the number of units sold by each salesperson. The salesperson who sells the most units wins the contest. Build a java application that inputs a series of 10 integers and determines and prints the largest integer. Your program should use at least the following three variables:
 - a. counter: A counter to count to 10 (i.e. to keep track of how many numbers have been input and to determine when all 10 numbers have been processed.
 - b. number: The inter most recently input by the user.
 - c. largest: The largest number found so far.

Note: Every time the sales figure of one employee is entered, the application should ask the user if they want to enter any more sales figures of a salesperson!
2. Write an application that prompts the user to enter the size of the side of a square, then displays a hollow square of that size made of asterisks. Your program should work for squares of all side lengths between 1 and 20.

3. Write a program to compute the following formula.

$$e = 1/0! + 1/1! + 1/2! + 1/3! + \dots + 1/n!$$

4. Using an enhanced for (for-each) loop, copy the content of one 3-dimensional array to another 3-dimensional array and display its contents.
5. Create the following vase pattern using a loop:

```
\      /
/      \
\      /
/      \
\      /
/      \
```



Object Oriented Programming using JAVA

(MCA292)

Assignment - 4

Inheritance and Dynamic Polymorphism

1. Create a general class ThreeDObject and derive the classes Box, Cube, Cylinder and Cone from it. The class ThreeDObject has methods wholeSurfaceArea() and volume(). Override these two methods in each of the derived classes to calculate the volume and whole surface area of each type of three-dimensional object. The dimensions of the objects are to be taken from the users and passed through the respective constructors of each derived class. Write a main method to test these classes.
2. Create a base class Building that stores the number of floors of a building, number of rooms and it's total footage. Create a derived class House that inherits Building and also stores the number of bedrooms and bathrooms. Demonstrate the working of the classes.
3. In the earlier program, create a second derived class Office that inherits Building and stores the number of telephones and tables. Now demonstrate the working of all three classes.
4. Create a base class Distance which stores the distance between two locations in miles and a method travelTime(). The method prints the time taken to cover the distance when the speed is 60 miles per hour. Now in a derived class DistanceMKS, override travelTime() so that it

prints the time assuming the distance is in kilometers and the speed is 100 km per second. Demonstrate the working of the classes.

5. Create a base class called “vehicle” that stores number of wheels and speed. Create the following derived classes –

- “car” that inherits “vehicle” and also stores number of passengers.

- “truck” that inherits “vehicle” and also stores the load limit.

Write a main function to create objects of these two derived classes and display all the information about “car” and “truck”. Also compare the speed of these two vehicles – car and truck and display which one is faster.



Object Oriented Programming using JAVA

(MCA292)

Assignment - 5

Abstract class & Interface in Java.

1. Design an abstract class having two methods. Create Rectangle and Triangle classes by inheriting the shape class and override the above methods to suitably implement for Rectangle and Triangle class.
2. Write a program to create a class named Vehicle having protected instance variables regnNumber, speed, colour, ownerName and a method showData () to show “This is a vehicle class”. Inherit the Vehicle class into subclasses named Bus and Car having individual private instance variables routeNumber in Bus and manufacturerName in Car and both of them having showData () method showing all details of Bus and Car respectively with the content of the super class’s showData () method.
3. Create an interface Department containing attributes deptName and deptHead. It also has abstract methods for printing the attributes. Create a class hostel containing hostelName, hostelLocation and numberOfRooms. The class contains methods for getting and printing the attributes. Then write a Student class extending the Hostel class and implementing the Department interface. This class contains attributes studentName, regdNo, electiveSubject and avgMarks. Write suitable getData and printData methods for this

class. Also, implement the abstract methods of the Department interface. Write a driver class to test the Student class. The program should be menu driven containing the options:

- i) Admit new student
- ii) Migrate a student
- iii) Display details of a student

For the third option, a search is to be made on the basis of the entered registration number.

4. Create an abstract class Accounts with the following details:

Data Members:

- (a) Balance
- (b) accountNumber
- (c) accountHoldersName
- (d) address

Methods:

- (a) withdrawl()- abstract
- (b) deposit()- abstract
- (c) display() to show the balance of the account number

Create a subclass of this class SavingsAccount and add the following details:

Data Members:

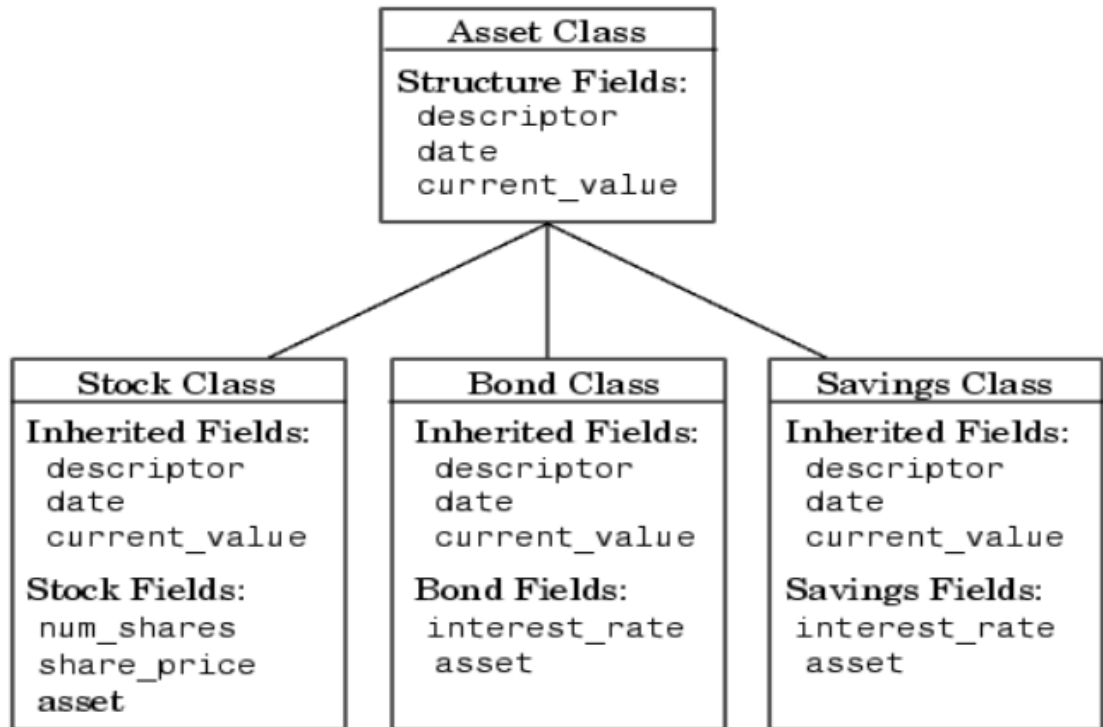
- (a) rateOfInterest

Methods:

- (a) calculateAount()

5. Implement the below Diagram.

Here, Asset class is an abstract class containing an abstract method displayDetails() method. Stock, bond and Savings class inherit the Asset class and displayDetails() method is defined in every class.





Object Oriented Programming using JAVA

(MCA292)

Assignment - 6 Exception Handling

1. Write a Java program using try and catch to generate Array Index Out of Bound Exception and Arithmetic Exception.
2. Write a class that keeps a running total of all characters passed to it (one at a time) and throws an exception if it is passed a non-alphabetic character.
3. Write a program that takes a value at the command line for which factorial is to be computed. The program must convert the string to its integer equivalent. There are three possible user input errors that can prevent the program from executing normally.
 - The first error is when the user provides no argument while executing the program and an `ArrayIndexOutOfBoundsException` is raised. You must write a catch block for this.
 - The second error is `NumberFormatException` that is raised in case the user provides a non-integer (float double) value at the command line.
 - The third error is `IllegalArgumentException`. This needs to be thrown manually if the value at the command line is 0.

4. Create a user-defined exception named CheckArgument to check the number of arguments passed through the command line. If the number of arguments is less than 5, throw the CheckArgumentexception, else print the addition of all the five numbers.
5. Write a java program to create a custom Exception that would handle at least 2 kinds of Arithmetic Exceptions while calculating a given equation (e.g. $X+Y*(P/Q)Z-I$).



Object Oriented Programming using JAVA

(MCA292)

Assignment - 8

Threads, Multithreading & Thread Synchronization

1. Write a Java program in which a total of 4 threads should run. Set different priorities for the thread.
2. Write a Java Program to Create a Thread that Implements the Runnable Interface.
3. Write a Java Program to Check the Priority Level of a Thread.
4. Write a Java Program Defining Thread By Extending the Thread class.
5. Write a Java Program to Get the Name of a Running Thread.
6. Write a Java Program to Stop a Thread.
7. Write a Java Program to Check Whether Define a Thread Class Without Defining run() Method in the Class.
8. Write a Java Program to Show that Method Will be Verified Whether it is Synchronized or Not.
9. Create 4 threads with priority 1,3,5,7 respectively. Update a counter in each of the threads for 10 ms. Print the final value of the count for each thread.
10. Write a Java Program to Use Method Level Synchronization.