Lab Assignments
on
Object Oriented Programming using Java

Assignment 1

Objective: To learn how to write simple Java programs.

Write programs to perform following tasks.

1. Print your name.
2. Read the price of an item in the decimal form (like 75.95) and print the output in paise (like 7595 paise).
3. Calculate the area and the perimeter of a rectangle.
4. Convert the given temperature in Fahrenheit to Celsius using the following conversion formula: $C = (F - 32)/1.8$.
5. Determine sum of the following series for given value of n: $1 + 1/2 + 1/3 + \cdots + 1/n$. Print the result up to two decimal places.
6. Take your first and last name as command line arguments and print after concatenating them into a string.
7. Find the product of the sum of digits and the integer with its digits reversed of a given integer (take input using command-line argument).
8. Find the factorial of a given integer number using recursion (take input using command-line argument).
9. Show Fibonacci series up to n-th terms using recursion.
10. Calculate the series $1 + x + x^2 + x^3 + x^4 + \cdots$ up to n-th terms).
11. Calculate the simple interest (S.I.) with inputs of principle (p), time in years (n) and rate of interest (r) [take input using command-line argument].
12. Find the real roots of the quadratic equation $ax^2 + bx + c = 0$ where a, b and c are constants.
13. Print all prime number within a given range.
14. Print all Armstrong number within a given range.
15. Calculate GCD and LCM of two numbers. (Implement 2 versions).
16. Find sum and average of several integers (in an array) using enhanced-for loop.
17. Implement linear search. Take the values via command line argument.
18. Implement binary search.
19. Implement Bubble sort, Selection sort, Insertion sort in a single menu driven program.
20. Count number of words in a string.
21. Check whether a string is palindrome or not. Make this Case Insensitive.
22. Perform addition, subtraction and multiplication of two matrices using a menu driven program.

Assignment 2

Objective: To learn about the concept of classes and objects.

Write programs to perform the following tasks.

1. Add two numbers by taking input using command-line input, `Scanner` class and `BufferedReader` class. Implement two versions.
2. Find surface area and volume of a cylinder using constructors - keyboard input or command-line input.
3. Find surface area and volume of cone using constructors - keyboard input or command-line input.
4. Add two complex numbers using concept of methods returning objects and methods taking objects as parameters.
5. Find a number from an array of numbers.
6. Show that `static` blocks execute before any object creation and implement the use of static variable.
7. Create a class; put a method inside this class which will return a class reference; return the same class and/or different class object.
8. Solve the problems below:
   a) Make a `Student` class with proper attributes like roll number, name, stream, and college. From `main()` create such two students and show their information.
   b) Consider the `Student` class in the previous program. Assume that a student studies 6 subjects. Each subject has a title, internal marks and theory marks. Write a Program to define the `Student` class, including the subjects as an array. From `main()`, create two such students and show their information including the names of the subjects and grand total marks.
   c) Consider the `Student` class in the first program 8.a. Assume that students study varying number of subjects. Each subject has a title, internal marks and theory marks. Write a Program to define Student class including the subjects as vararg argument of constructor. From main create such two students and show their information including subjects' name and grand total marks.
9. Design a class to represent a bank account. Include the following:

   Fields
   - Name of the depositor
   - Address of the depositor
   - Account number
   - Balance amount in the account

   Methods
   - To assign initial values
   - To deposit an amount
   - To withdraw an amount after checking balance
   - To display the name, address and balance of a customer.

   From `main()` create object and call these methods.

Assignment 3

Objective: To learn about inheritance, polymorphism, and abstract classes.

1. Create a class shape with three methods to calculate area of triangle, rectangle and square with method overloading.
2. Create an abstract class `Shape` with two abstract methods, `area()` and `disp()`. Now design three concrete classes `Rectangle`, `Circle` and `Triangle` can compute area and display its separately.
3. Overload the constructors for classes `Area` and `Volume` of a rectangular figure and also display its area and volume. `Area` is the superclass and `Volume` is the subclass.
4. Create a class `Employee`, having instance variables `name` and `id`. Create its subclass named `Scientist` which has instance variables `no_of_publication` and `experience`. Now create its subclass, say `DScientist` which has instance variable `award`. Put a method like: `public String toString() { }` in every class where you describe about the class and from `main()` method create object of each class and print each object.
5. Create a class with a method `void show()` and make three subclasses of it and all subclasses have this `show()` method overridden and call those methods using their corresponding object references.
6. Do the problem 4 using dynamic method dispatching.
7. Assume that a bank maintains two kinds of account for its customers, one called savings account and other called current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance (say Rs. 1000) and if the balance falls below this level a service charge is imposed (say Rs. 100). Create a class `Account` that stores customer name, account number and type of account. From this class derive two classes `Curr_Acct` and `Savn_Acct` respectively to make them more specific to their requirements. Include the necessary methods to achieve the following tasks:
   a) Accept deposit from a customer and update the balance.
   b) Display the balance.
   c) Compute and deposit interest.
   d) Permit withdrawal and update the balance.
   e) Check for minimum balance, impose penalty, if necessary, and update balance.

   Use constructors to initialise the class members.
8. Create a class `Parent` having instance variables `id`, `name` and `address`. Create a class `ChildOne` having instance variables `id`, `name`, `address` and `marks`. Also create another class `ChildTwo` with instance variables `id`, `name`, `address`, `qualification` and `salary`. Within each class define your own method to display values of these variables. Design the program using `super` call with proper parameter and use object of each class from `main()` to display their properties.

Assignment 4

Objective: To learn the concept of interfaces, inner classes and packages.

1. Create an interface named `Shape`. Create two subclasses of it named `Circle` and `Sphere`. Create objects of the two classes and calculate their area/surface area.
2. Create an interface named `CircularBase` (that contains base details). Create another class `_3dShape` (Contains height, volume). Inherit two classes `Cone` and `Cylinder` from the interface and the class.
3. Create a class which contains an inner class. Show that inner class can use member of outer class directly, but Outer class can use member of inner class only through its object. Check the name of class file, you created.
4. Create two interfaces, each with two methods. Inherit a new interface from the two, adding a new method. Create a class by implementing the new interface and also inheriting from a concrete class. In `main()` method, create an object of derived class and call the methods [do all without package statement].
5. Create a class with variable(s) and method(s) (all will be default accessed) under package `pOne`. Now create a class under package `pTwo`, which is subclass of firstly created class. In the method here (*i.e.* class of `pTwo`) call variable(s) and method(s) of previous class (*i.e.* class of `pOne`). If errors come, rectify them. Now from `main()` (under working directory) access members of the second class.
6. Create an interface containing three methods, in a package `pkgOne`. Implement the interface from a class under package `pkgTwo`. From `main()`, under working directory, create object of the class and call methods of interface.

Assignment 5

Objective: To learn about `String`, `Collection` and `File`.

1. Take a sting from keyboard and convert into character array (new one).
2. Take a string from keyboard and a `char` array (filled up to length 5). Now append the string to that `char` array. Show the `char` array.
3. Write a java code to differentiate `equals()` method and `==` operator.
4. Find length of a string taken from keyboard and also find the length of that string except the spaces at the beginning and the end of the string.
5. Sort ten names in ascending order.
6. Check if `"Tech"` is present in `"University of Technology"` or not. If yes return its position.
7. Take a sentence and convert it into string arrays and sort the words using any sorting technique.
8. Show that the `String` objects are immutable but `StringBuffer` objects are mutable.
9. Convert a `StringBuffer` object into a `String` object. Print the final result.
10. Check whether a given string is a palindrome or not. Ignore the cases.
11. Convert a string into an array of strings and display them [use command-line argument].
12. Take a shopping list of five items from the command line and store them in a vector.
13. Write a program to concatenate the contents of two strings.