

Assignment – 16 (Multi-Level Inheritance)

1. Create a class *Number* :
Data member: An array of type integer.
Constructor: Constructor with one parameter n, that is the size of the array.
Allocate n memory for the array and input n numbers into the array.
Method-1: To display all the values in the array.
Derive a class *OddNum* from the class *Number*:
Data member: An array of type integer.
Constructor: To count the odd numbers present in the array of its base class *Number* and accordingly allocate memory for its own array.
Method-1: To copy all odd numbers from its base class array to its own array.
Method-2: To display all odd numbers.
Derive a class *PrimeNum* from the class *OddNum*:
Data member: An array of type integer.
Constructor: To count the prime numbers present in the array of its base class *OddNum* and accordingly allocate memory for its own array.
Method-1: To copy all prime numbers from its base class array to its own array.
Method-2: To display all prime numbers.
2. Define a class ***Employee*** having private members – id, name, department, salary. Define default and parameterized constructors. Create a subclass called “***Manager***” with private member bonus. Define methods ***accept()*** and ***display()*** in both the classes. Create *n* objects of the ***Manager*** class and display the details of the manager having the maximum total salary (salary+bonus).

Assignment -17

(String Operations)

1. Write a java program to define a class **UserString** and to perform the following operations using different methods.
 - a) Count all the characters
 - b) Count no of words
 - c) Compare two strings
 - d) Convert to uppercase
 - e) Convert to lowercase
 - f) Concatenate two strings
 - g) Check a string is palindrome or not
 - h) Find the position of a given character
 - i) Make a substring from a desired start and end position.
 - j) Search the presence of a substring.
 - k) Replace a substring with a new string.
 - l) Swap two substrings between two strings.

Assignment-18

(Abstract Class)

- Q1. Create a class Shape having data members length, breadth, height and abstract methods such as volume and surfaceArea. Inherit this class into cube, cylinder and cuboid classes. Redefine the required methods to calculate and display the volume and surface area of each shape.
- Q2. Design an abstract class fruit with data members colour, taste and an abstract method display. Inherit this class to other classes such as Apple, Banana, Orange and Strawberry. Redefine the display method to show the color and taste of each fruit along with its name.

Assignment -19

(Interface)

1. Define an interface “IntOperations” with methods to check whether a number is positive/negative, even/odd, prime, palindrome and operations like factorial and sum of digits. Define a class MyNumber having one private data member of type int. Write a default constructor to initialize it to 0 and another constructor to initialize it to a value (Use this). Implement the above interface. Create an object in main method. Input a number and write a menu driven program to check different properties of the number using above methods.
2. Define an interface “StackOperations” which declares methods for a static stack. Define a class “MyStack” which contains an array and top as data members and implements the above interface. Initialize the stack using a constructor. Write a menu driven program to perform all operations (Push, POP, Peak) on a MyStack object.

Assignment -20

(Package)

1. Create a package named **mathop**. Define class MathsOperations with static methods to find the maximum and minimum of n numbers. Create another package **statop**. Define class StatsOperations with methods to find the average and median of n numbers. Import these packages to use the above methods to perform above operations on n numbers.
2. Create a package called **nodepack** which contains the class “Node”. Create another package called **listpack** which contains the class “LinkedList” representing methods to create a Single Linked list, Add a node to the list and traverse the list. Write a menu driven program in main to create a Single Linked list, Add nodes and display the List. The elements are passed as user input.

Assignment – 21

(Built-In-Exception)

1. Input two numbers as numerator and denominator for division. Write a program to show an `ArithmeticException` if the division is not possible when denominator is 0.
2. Define an array of size `n` and set some values to it. Show an `ArrayIndexOutOfBoundsException` when trying to access the index that is more than size of the array.
3. Write a program to show the use of `NullPointerException`.

Assignment – 22

(UserDefined-Exception)

1. Define Exceptions `VowelException`, `BlankException`, `ExitException` to restrict the input of vowel, space and 'X'. Write another class `TestException` which reads a character from command line. If it is a vowel, throw `VowelException`, if it is a blank space throw `BlankException` and for a character 'X' throw an `ExitException` and terminate the program. For any other character, display "Valid character".
2. Write a program which accepts two integers and an arithmetic operator from the command line and performs the operation. Check the following user defined exceptions:
 - i. If the number of arguments are less than 3 then throw "`FewArgumentsException`".
 - ii. If the operator is not an Arithmetic operator, throw "`InvalidOperatorException`".
 - iii. If result is -ve, then throw "`NegativeResult`" exception.
3. Create a class **Student** with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not between 15 and 21 then generate user-defined exception "`InvalidAgeException`". If name contains numbers or special characters raise exception "`InvalidNameException`". Define the two exception classes.
4. Define class `MyDate` with members day, month and year. Define default and parameterized constructors. Accept values from the command line and create a date object. Throw user defined exceptions – "`InvalidDayException`" or "`InvalidMonthException`" if the day or month are invalid. If the date is valid, display message "Valid Date".