**Java Program List**

1. Write a program to check whether a number is prime.
2. Write a program to print a Fibonacci series upto a limit.
3. Write a program to check whether a number is even or odd.
4. Write a program to check whether an year is leap year.
5. Java program to remove duplicate elements from an array.
6. Java program to create a matrix and fill it with prime numbers.
7. Java program to check whether a given matrix is Lower Triangular Matrix or not
8. Java program to check Given String is Palindrome String or not in Java.
9. Java program to get string and count number of words in a provided string.
10. Java Program to divide a string in 'N' equal parts.
11. Design a class to represent a bank account. Include the following members:

Data members: Name of the depositor, account no, type of account and balance amount

Methods: to deposit an amount. To withdraw an amount after checking balance, to display the name and balance

Use constructors to provide the initial values.

1. Create a linked list of n nodes and then reverse the order of nodes.
2. WAP to increment the employee salaries on the basis of their designation(Manager-5000, General Manager-10000, CEO-20000, worker-2000). Use employee name, id, designation , salary as data member and inc\_sal as member function.
3. Assume that a bank maintains two kinds of account for its customers, one called as savings account and the other as current account. The saving 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 and if the balance falls below this level, a service charge is imposed. Class account stores customer name, account number and the type of account.

Include member functions:-

* Accept deposit from a customer and update the balance
* Display the balance
* Compute and deposit interest
* Permit withdrawal and update balance
* Check for minimum balance, impose penalty and update the balance

1. Design three classes: Student, Exam and Result. The student class has data members such as roll no, name etc. Create a class Exam by inheriting the Student class. The Exam class adds data members representing the marks scored in six subjects. Derive the Result from class Exam and it has its own members such as total marks and average. Calculate the total marks and average.
2. Write a program with given interfaces MotorBike and Cycle, then implement in child class TwoWheeler and display distance & speed.
3. An interface called RegularPolygon with two abstract methods: getNumSides and getSideLength.

A class EquilateralTriangle that implements the interface, has getNumSides return 3 and getSideLength return an instance variable that is set by the constructor.

A class Square that implements the interface, has getNumSides return 4 and getSideLength return an instance variable that is set by the constructor.

Add a static totalSides method, that given a RegularPolygon[], returns the sum of the number of sides of all the elements.

Add method:

getPerimeter (n \* length, where n is the number of sides)

1. Create a class Employee. Derive 3 classes from this class namely, Programmer, Analyst & Project Leader. Take attributes and operations on your own.
2. Create a class with a **main( )** that throws an object of class **Exception** inside a **try** block. Give the constructor for **Exception** a **String** argument. Catch the exception inside a **catch** clause and print the **String** argument. Add a **finally** clause and print a message to prove you were there.
3. Create a program to ask the user for a real number and display its square root. Errors must be trapped using "try..catch"
4. Create a try block that is likely to generate three types of exception and then incorporate necessary catch blocks to catch and handle them appropriately.
5. Create a class MyClass and create three methods myMethod1(), Method2() and Method3(). Invoke Method2() from Method1() and Method3() from Method2(). Write a code that can throw an exception inside myMethod3()
6. Write a program to deposit cash, withdraw in a bank using multithreading.
7. Implement three classes: Storage, Counter, and Printer. The Storage class should store an integer. The Counter class should create a thread that starts counting from 0 (0, 1, 2, 3 ...) and stores each value in the Storage class. The Printer class should create a thread that keeps reading the value in the Storage class and printing it.

Write a program that creates an instance of the Storage class and sets up a Counter and a Printer object to operate on it.

1. Create a library system with methods for returning and issuing a book. Apply multithreading synchronization concept and exception handling.