1. Implement a java program accept command line arguments and display same.

PASSING A SINGLE STRING THROUGH ARGUMENT (COMMAND LINE)

```
public class string {
  public static void main(String[] args) {
    if(args.length==0) {
      System.out.println("No argument provided");
    }
    else {
      String name = args[0];
      System.out.println("Hello " + args[0]);
    }
  }
}
```

PASSING INTEGERS TO CARRY OUT OPERATIONS THROUGH ARGUMENT (COMMAND LINE)

```
public class operation {
  public static void main(String[] args) {
    if(args.length==0) {
      System.out.println("No argument provided");
    }
    else {
      int sum = 0;
      for(int i = 0; i<args.length; i++) {
        System.out.println("The arguments provided are: "
         + args[i]);
         sum += Integer.parseInt(args[i]);
      }
      System.out.println("Sum is: " + sum);
    }
  }
}
```

2. Implement a java program add two numbers by accepting user input using BufferReader.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class AddTwoNumbers {
    public static void main(String[] args) {
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

        try {
            // Accept the first number from the user
            System.out.print("Enter the first number: ");
            String input1 = reader.readLine();
            double number1 = Double.parseDouble(input1);

            // Accept the second number from the user
            System.out.print("Enter the second number: ");
            String input2 = reader.readLine();
            double number2 = Double.parseDouble(input2);

            // Add the two numbers
            double sum = number1 + number2;

            // Display the result
            System.out.println("The sum of " + number1 + " and " + number2 + " is: " + sum);
        } catch (IOException | NumberFormatException e) {
            System.out.println("Error: Invalid input. Please enter valid numbers.");
        } finally {
            try {
                reader.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

3. Implemented Java programme to take values of length and breadth of rectangle from user and check if it is square or not? If it is a square, find the area of square.

```
import java.util.Scanner;

class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter the length: ");
      int length = sc.nextInt();
      System.out.print("Enter the breadth: ");
      int breadth = sc.nextInt();
      if(length==breadth) {
         System.out.println("It is a square!");
         int area = length*breadth;
         System.out.println("Area is: " + area);
      }
      else {
         System.out.println("It is a rectangle");
      }
   }
}
```

4. Write a program to print the names of Students by creating a Student class. If no name is passed while creating an object of Student class, then the name should be “Unknown”, otherwise the name should be equal to the String value passed while creating object of Student class.

```
class Student {
   String name;
   int id;
   int age;

   public Student(String name, int id, int age) {
      this.name = (name != null && !name.isEmpty()) ? name : "Unknown";
      this.id = id;
      this.age = age;
      System.out.println(this.name);
   }

   public static void main(String args[]) {
      Student s = new Student("gfgv", 9, 9);
      Student S1 = new Student("", 0, 0);
   }
}
```

5. Cerate a class ‘Student’ with three data members which are name, age and address. The constructor of the class assigns default values name as “unknown”, age as ‘0’ and address as “not available”. It has two members with the same name as ‘setInfo’. First method has two parameters for name and age and assigns the same whereas the second method takes has three parameters which are assigned to name, age and address respectively. Print the name, age and address of 10 student.

```
public class Student {
  private String name;
  private int age;
  private String address;

  // Constructor with default values
  public Student() {
    this.name = "unknown";    this.age = 0;  this.address = "not available";
  }

  public void setInfo(String name, int age) {
    this.name = name;
    this.age = age;
  }

  public void setInfo(String name, int age, String address) {
    this.name = name;
    this.age = age;
    this.address = address;
  }

  // Method to print student information
  public void printInfo() {
    System.out.println("Name: " + name);
    System.out.println("Age: " + age);
    System.out.println("Address: " + address);
  }

  public static void main(String[] args) {
    Student[] students = new Student[10];

    // Create and set information for each student
    for (int i = 0; i < 9; i++) {
      students[i] = new Student();
      students[i].setInfo("Student " + (i + 1), 20 + i, "Address" + (i + 1));
    }

    // Print information of each student
    for (int i = 0; i < 9; i++) {
      System.out.println("Student " + (i + 1) + " information:");
      students[i].printInfo();
      System.out.println();
    }
  }
}
```

6. Implement a java program that demonstrates ArrayList.

```
import java.util.*;
class Collection1{
  public static void main(String args[]){
    ArrayList<String> list=new ArrayList<String>();//Creating arraylist
    list.add("Anushka");//Adding object in arraylist
    list.add("Tanisha");
    list.add("Parth");
    list.add("Ameya");
    //Traversing list through Iterator
    Iterator itr=list.iterator();
    while(itr.hasNext()){
    System.out.println(itr.next());
    }
  }
}
```

7. Implement a java program that demonstrates Vector.

```
import java.util.*;
public class Collection3 {
    public static void main(String args[]){
        Vector<String> v=new Vector<String>();
        v.add("Ayush");
        v.add("Amit");
        v.add("Ashish");
        v.add("Garima");
        Iterator<String> itr=v.iterator();
        while(itr.hasNext()) {
            System.out.println(itr.next());
        }
    }
}
```

8. Create separate Engine, Tyre and Door class. Create a car class using these classes and show functionality of each component in the car.

```
abstract class Ab {
        abstract void show();
}

class Engine extends Ab {
        void show() {
                System.out.println("Sound proof Engine");
        }
}

class Tyre extends Ab {
        void show() {
                System.out.println("Nice Tyres");
        }
}

class Door extends Ab {
        void show() {
                System.out.println("Built Quality is great");
        }
}

public class car {
        public static void main(String[] args) {
                Ab A1 = new Engine();
                Ab A2 = new Tyre();
                Ab A3 = new Door();
                A1.show();
                A2.show();
                A3.show();
        }
}
```

9. Create Vehicle interface with name, maxPassenger and maxSpeed variables. Create LandVehicle and SeaVehicle Interface from Vehicle interface, LandVehicle has numWheels variable and drive method, SeaVehicle has displacement variable and launch method. Create Car class from LandVehicle, HoverCraft from LandVehicle and SeaVehicle interface. Also create Ship from SeaVehicle. Provide additional methods in HoverCraft as enterLand and enterSea. Similarly provide other methods for class Car and Ship Demonstrate all classes in a application.

```
interface Vehicle {
   String name = "";
   int maxPassenger = 0;  int maxSpeed = 0;
}
interface LandVehicle extends Vehicle {
   int numWheels = 0;
   void drive();
}

interface SeaVehicle extends Vehicle {
   int displacement = 0;
   void launch();
}

class Car implements LandVehicle {
   String name;
   int maxPassenger, maxSpeed,
numWheels;

   public Car(String name, int maxPassenger,
int maxSpeed, int numWheels) {
      this.name = name;
      this.maxPassenger = maxPassenger;
      this.maxSpeed = maxSpeed;
      this.numWheels = numWheels;
   }

   public void drive() {
      System.out.println("Car is being
driven.");
   }
}

class HoverCraft implements LandVehicle,
SeaVehicle {
   String name;
   int maxPassenger, maxSpeed,
numWheels, displacement;

   public HoverCraft(String name, int
maxPassenger, int maxSpeed, int
numWheels, int displacement) {
      this.name = name;
      this.maxPassenger = maxPassenger;
      this.maxSpeed = maxSpeed;
      this.numWheels = numWheels;
      this.displacement = displacement;
   }

   public void drive() {
      System.out.println("HoverCraft is being
driven on land.");
   }

   public void launch() {
      System.out.println("HoverCraft is being
launched into sea.");
   }
}

class Ship implements SeaVehicle {
   String name;
   int maxPassenger, maxSpeed;
   int displacement;

   public Ship(String name, int
maxPassenger, int maxSpeed, int
displacement) {
      this.name = name;
      this.maxPassenger = maxPassenger;
      this.maxSpeed = maxSpeed;
      this.displacement = displacement;
   }

   public void launch() {
      System.out.println("Ship is being
launched into sea.");
   }
}

 class VehicleDemo {
   public static void main(String[] args) {
      Car myCar = new Car("Sedan", 5, 200,
4);
      HoverCraft myHoverCraft = new
HoverCraft("HoverCraft", 10, 150, 0, 1000);
      Ship myShip = new Ship("Cargo Ship",
50, 50, 5000);
      myCar.drive();
      myHoverCraft.drive();
      myHoverCraft.launch();
      myShip.launch();
   }
```

10. Develop a mathematical package for Statistical operations like Mean, Median, Average, Standard deviation. Create a sub package in the math package – convert. In “convert” package provide classes to convert decimal to octal, binary, hex and vice-versa. Develop application program to use this package and build executable jar file of it.

```
package mypack;
import java.util.Arrays;
public class pack1 {
    public static double mean(double[] numbers) {
        double sum = 0;
        for (double num : numbers) {
        sum += num;
    }
    return sum / numbers.length;
}
    public static double median(double[] numbers) {
        Arrays.sort(numbers);
        int middle = numbers.length / 2;
        if (numbers.length % 2 == 0) {
            return (numbers[middle - 1] + numbers[middle]) / 2.0;
        }
        else {
            return numbers[middle];
        }
}
    public static double average(double[] numbers) {
        return mean(numbers);
    }
}
```

MAIN PROGRAM:

```
import mypack.pack1;
class Test {
    public static void main(String[] args) {
        double[] d = { 20, 50, 60, 80 };
        pack1 m = new pack1(); // Create an instance of pack1
        System.out.println("Mean is " + m.mean(d)); // Call methods on pack1 instance
        System.out.println("Median is " + m.median(d));
        System.out.println("Average is " + m.average(d));
    }
}
```

```
javac -d . pack1.java

javac test.java

javac test
```

11. Write the following programs for an array of integer data to be initialized. During the initialization if a user enters a value other than integer value, then it will throw InputMismatchException exception. On the occurrence of such an exception, your program should print “You entered bad data”. If there is no such exception it will print the total sum of the array.

	Input	Output
Test case 1	2 1 1.0	You entered bad data.
Test case 2	3 5 2 1	8
Test case 3	2 1 j	You entered bad data.

```
import java.util.*;
public class Test1 {
   public static void main(String[] args) {
      int total = 0;
      Scanner sc = new Scanner(System.in);

      try {
         System.out.print("Enter size of array: ");
         int size = sc.nextInt();
         int a[] = new int[size];
         System.out.println("Enter Numbers:");
         for (int i = 0; i < size; i++) {
            a[i] = sc.nextInt();
         }
         for (int i = 0; i < size; i++) {
            total += a[i];
         }
         System.out.println("Total is " + total);
      }

      catch (Exception e) {
         System.out.println("You Entered Bad Data ");
      }
   }
}
```

12. In the following program, there may be multiple exceptions. You have to complete the code using only try-catch block to handle all the possible exceptions. For example, If users input is 1, then it will throw and catch “java.lang.NullPointerException”.

	Input	Output
Test Case 1	1	java.lang.NullPointerException
Test Case 2	50	No exception
Test Case 3	5	No exception
Test Case 4	0	java.lang.ArithmeticException: / by zero

```
import java.util.*;

public class Test3 {
    public static void main(String[] args) {
        int a;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Value: ");

        try {
            a = sc.nextInt();
            if (a == 1) {
                throw new NullPointerException();
            } else if (a == 2) {
                throw new ArrayIndexOutOfBoundsException();
            } else if (a == 3) {
                throw new ArithmeticException();
            } else {
                System.out.println("No Exception");
            }
        }
        catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

13. Implement a java program based on String. (Length, CharAt, CompareTo, Trim, Replace all)

```
import java.util.*;
public class S1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String a, b;
        int a1;
        System.out.println("Enter Two Strings: ");
        a = sc.nextLine();
        b = sc.nextLine();
        S2 s = new S2();
        do {
            System.out.println("1. Length\n2.
CharAt\n3. CompareTo\n4. Concat\n5. Upper
case\n6. Lower case\n7. Trim\n8. Replace\n9.
Replace all\n10.EndsWith\n11.StartsWith ");
            System.out.print("Enter Your Choice: ");
            int i = sc.nextInt();
            switch (i) {
                case 1: s.T1(a); break;
                case 2: s.T2(a); break;
                case 3: s.T3(a, b); break;
                case 4: s.T4(a, b); break;
                case 5: s.T5(a); break;
                case 6: s.T6(a); break;
                case 7: s.T7(a); break;
                case 8: s.T8(b); break;
                case 9: s.T9(b); break;
                case 10: s.T10(a); break;
                case 11: s.T11(a); break;
                default: System.out.println("Please Give
Right choice"); break;
            }
            System.out.print("Do You Want to Continue
(1 for Yes and 0 for No) : ");
            a1 = sc.nextInt();
        } while (a1 == 1);
    }
}

class S2 {
    void T1(String a) {
        System.out.println("length is: " + a.length()); }
    void T2(String a) {
        System.out.println(a.charAt(0)); }
    void T3(String a, String b) {
        System.out.println(a.compareTo(b)); }
    void T4(String a, String b) {
        System.out.println(a.concat(b)); }
    void T5(String a) {
        System.out.println(a.toUpperCase()); }
    void T6(String a) {
        System.out.println(a.toLowerCase()); }
    void T7(String a) {
        System.out.println(a.trim()); }
    void T8(String b) {
        System.out.println(b.replace("sgu", "SGU")); }
    void T9(String b) {
        System.out.println(b.replaceAll("sgu", "SGU"));
}
    void T10(String a) {
        System.out.println(a.endsWith("va")); }
    void T11(String a) {
        System.out.println(a.startsWith("Ja"));}
}
```

14. Implement a java program to demonstrate use of static, non Static, this keywords. (On class, variable, method)

```
class StaticNonStatic {
        static int a=10;
        int b=20;

        static void staticMethod()      {
                System.out.println("This is Static method");
        }

        void nonstaticMethod() {
                System.out.println("This is nonStaticmethod");
        }

        void add(int a, int b) {
                this.a= a;
                this.b= b;
                int c=a+b;
                System.out.println("Add is:"+c);
        }

        public static void main(String args[]) {
                staticMethod();
                StaticNonStatic q14=new StaticNonStatic();
                q14.nonstaticMethod();
                q14.add(30,40);
        }
}
```

15. Implement a java program to demonstrate use of Super keyword. (On class, variable, method)

```
class A {
    int i, j;
    public A(int a, int b) {
        i = a;
        j = b;
    }
    void show() {
        System.out.println("i=" + i + "\n" + "j=" + j);
    }
}

class B extends A {
    int k;
    B(int a, int b, int c) {
        super(a, b);
        k = c;
    }

    void show() {
        super.show(); // Accessing superclass method
        System.out.println("k=" + k);
    }

    void accessSuperVariable() {
        System.out.println("Parent Variable from Child: " + super.i); // Accessing superclass variable
    }
}

public class Main {
    public static void main(String[] args) {
        B b = new B(2, 3, 4);
        b.show();
        b.accessSuperVariable();
    }
}
```

16. Implement a java program to demonstrate use of final keyword. (On class, variable, method)

```
final class undefined {
  final int dim1,dim2;
  undefined(int x , int y) {
    dim1=x;
    dim2=y;
  }
   final int show();
}

class Rectangle extends undefined {
  int area,a,b;
  Rectangle(int a, int b) {
    super(a,b);
  }
  int show() {
    area= a*b;
    System.out.println("Area of Rectangle is " + (dim1*dim2));
    return (dim1*dim2);
  }
}

class Mainn {
  public static void main(String [] args) {
    Rectangle r = new Rectangle(10,20);
    r.show();
  }
}
```

17. Implement a java program to demonstrate Multithreading which implements Runnable class and extends thread class.

```
1.  class MultithreadingDemo extends Thread {
  public void run() {
    try {
      // Displaying the thread that is running
      System.out.println(
        "Thread " + Thread.currentThread().getId()
        + " is running" + "\nPriority is:" + Thread.currentThread().getPriority()
      );
    } catch (Exception e) {
      // Throwing an exception
      System.out.println("Exception is caught");
    }
  }
}

public class Multithread {
  public static void main(String[] args) {
    int n = 5; // Number of threads
    for (int i = 0; i < n; i++) {
      MultithreadingDemo object = new MultithreadingDemo();
      object.start();
    }
  }
}
```

```
2.  class Multithreading extends Thread {
  public void run() {
    try {
      System.out.println("Thread" + Thread.currentThread().getId() + "\nPriority is:" +
Thread.currentThread().getPriority());
    } catch (Exception e) {
      System.out.println("Exception is caught");
    }
  }
}
public class Main {
  public static void main(String[] args) {
    Multithreading m = new Multithreading();
    m.start();
    m.setPriority(Thread.MAX_PRIORITY);
    System.out.println("Max Priority:" + m.getState());

    Multithreading m1 = new Multithreading();
    m1.start();
    m1.setPriority(Thread.MIN_PRIORITY);
    System.out.println("Min Priority:" + m1.getState());
  }
}
```

18. Write a java program and create Registration form using Swing.

```
import javax.swing.*;
import java.awt.*;
class Sample {
  public static void main(String args[]) {
    JFrame f = new JFrame("Registration form!");
    f.setSize(800, 400);
    JPanel p = new JPanel();
    JLabel j1 = new JLabel("First name: ");
    JTextField t1 = new JTextField(30);
    JLabel j2 = new JLabel("Last name: ");
    JTextField t2 = new JTextField(30);
    JLabel j3 = new JLabel("Gender");
    JRadioButton r1 = new JRadioButton("MALE");
    JRadioButton r2 = new JRadioButton("FEMALE");
    ButtonGroup bg = new ButtonGroup();
    JLabel j4 = new JLabel("Address");
    JTextArea t4 = new JTextArea(3, 30);
    bg.add(r1);
    bg.add(r2);
    p.add(j1);
    p.add(t1);
    p.add(j2);
    p.add(t2);
    p.add(j3);
    p.add(r1);
    p.add(r2);
    p.add(j4);
    p.add(t4);
    JButton s = new JButton("Submit");
    p.add(s);
    f.add(p, BorderLayout.CENTER); // Adding panel to the center of the frame
    f.setVisible(true);
  }
}
```

19. Implement a java program to create JDBC connectivity with MySQL and perform operations.

```
import java.sql.*;
import java.util.*;

class MysqlCon2 {
    public static void main(String args[]) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/pradnya", "root", "");
            Statement stmt = con.createStatement();
            Scanner p = new Scanner(System.in);
            System.out.println("Enter name");
            String a = p.next();
            System.out.println("Enter Password");
            String b = p.next();
            String query = "insert into magennavar values('" + a + "','" + b + "')";
            int i = stmt.executeUpdate(query);
            if (i > 0) {
                System.out.println("Record added successfully");
            } else {
                System.out.println("Error");
            }
            String query1 = "select * from magennavar";
            ResultSet rs = stmt.executeQuery(query1);
            while (rs.next()) {
                System.out.println(rs.getString(1) + " " + rs.getString(2));
            }
            con.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```