❖Question:-1

WAP to find the average and sum of the N numbers Using Command line argument.

❖Answer:-

```
public class SumAndAverage {
    public static void main(String[] args) {
        int sum = 0;
        for (String arg : args) {
            sum += Integer.parseInt(arg);
        }
        double average = (double) sum / args.length;
        System.out.println("Sum: " + sum);
        System.out.println("Average: " + average);
}
 }
```

- Output:-

```
java SumAndAverage 10 20 30 40 50
Sum: 150
Average: 30.0
```

❖Question:-2

WAP to Demonstrate Type Casting.

❖Answer:-

```
public class TypeCasting {
   public static void main(String[] args) {
      // Implicit casting
      int i = 100;
      double d = i;
      System.out.println("Implicit casting (int to double)= " + d);
      // Explicit casting
      double x = 9.78;
      int y = (int) x;
      System.out.println("Explicit casting (double to int)= " + y);
}
}
```

- Output:-

Implicit casting (int to double)= 100.0

Explicit casting (double to int)= 9

❖Question:-3

WAP to find the number of arguments provide at runtime.

❖Answer:-

```
public class ArgumentCount {
    public static void main(String[] args) {
        System.out.println("Number of arguments: " +
args.length);
    }
}
```

- Output:-

```
java ArgumentCount arg1 arg2 arg3

Number of arguments: 3
```

❖Question:-4

WAP to Test the Prime number.

❖Answer:-

```
public class PrimeCheck {
   public static void main(String[] args) {
      int num = Integer.parseInt(args[0]);
      boolean isPrime = true;
      if (num <= 1) {
         isPrime = false;
      } else {
         for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
               isPrime = false;
               break;
            }
         }
      }
      if (isPrime) {
         System.out.println(num + " is a prime number.");
      } else {
         System.out.println(num + " is not a prime number.");
      }
   }
}
```

- Output:-

java PrimeCheck 29

29 is a prime number.

❖Question:-5

WAP to find the factorial of a given number using Recursion.

❖Answer:-

```
public class Factorial {
    public static void main(String[] args) {
        int num = Integer.parseInt(args[0]);
        System.out.println("Factorial of " + num + " is " + factorial(num));
    }

    public static int factorial(int n) {
        if (n == 0) {
            return 1;
        } else {
            return n * factorial(n - 1);
        }
}
}
```

- Output:-

```
java Factorial 5
Factorial of 5 is 120
```

❖Question:-6

WAP to design a class using abstract Methods and Classes.

❖Answer:-

```
abstract class Shape {
   abstract void draw();
}
class Circle extends Shape {
   void draw() {
      System.out.println("Drawing Circle");
   }
}
class Rectangle extends Shape {
   void draw() {
      System.out.println("Drawing Rectangle");
   }
}
public class TestShape {
   public static void main(String[] args) {
      Shape s1 = new Circle();
      Shape s2 = new Rectangle();
      s1.draw();
      s2.draw();
   }
}
```

- Output:-

```
Drawing Circle

Drawing Rectangle
```

❖Question:-7

WAP to design a String class that perform String Method(Equal,Reverse the string,change case).

❖Answer:-

```
public class StringManipulation {
   public static void main(String[] args) {
      String str = "HelloWorld";

      // Check if two strings are equal
      String str2 = "HelloWorld";
      System.out.println("Strings are equal: " + str.equals(str2));

      // Reverse the string
      String reversed = new StringBuilder(str).reverse().toString();
      System.out.println("Reversed string: " + reversed);

      // Change case
      String upperCase = str.toUpperCase();
      String lowerCase = str.toLowerCase();
      System.out.println("Upper case: " + upperCase);
      System.out.println("Lower case: " + lowerCase);
   }
}
```

- Output:-

```
Strings are equal: true
Reversed string: dlroWolleH
Upper case: HELLOWORLD
Lower case: helloworld
```

❖Question:-8

WAP to handle the Exception using try and multiple catch block.

❖Answer:-

```
public class MultipleCatch {
    public static void main(String[] args) {
        try {
            int a = Integer.parseInt(args[0]);
            int b = Integer.parseInt(args[1]);
            int result = a / b;
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Error: Division by zero");
        } catch (NumberFormatException e) {
            System.out.println("Error: Number format exception");
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error: Insufficient arguments");   }  } }
```

- Output:-

  java MultipleCatch 10 0

  Error: Division by zero

❖Question:-9

WAP that Implement the Nested try Statements.

❖Answer:-

```
public class NestedTry {
   public static void main(String[] args) {
      try {
         int[] arr = new int[5];
         try {
            arr[5] = 30 / 0;
         } catch (ArithmeticException e) {
            System.out.println("Arithmetic exception caught");
         }
         arr[6] = 10;
      } catch (ArrayIndexOutOfBoundsException e) {
         System.out.println("Array index out of bounds exception
caught");
      }
   }
}
```

- Output:-

Arithmetic exception caught

Array index out of bounds exception caught

❖Question:-10

WAP to create a package that access the member of external class as well as same package.

❖Answer:-

```
// ExternalClass.java
package mypackage;
public class ExternalClass {
   public void display() {
      System.out.println("Hello from ExternalClass");
   }     }
// InternalClass.java
package mypackage;
public class InternalClass {
   public void display() {
      System.out.println("Hello from InternalClass");
   }   }
import mypackage.ExternalClass;
import mypackage.InternalClass;
public class TestPackage {
   public static void main(String[] args) {
      ExternalClass ext = new ExternalClass();
      InternalClass intl = new InternalClass();
      ext.display();
      intl.display();
   } }
```

- Output:-

Hello from ExternalClass

Hello from InternalClass

❖Question:-11

WAP that import the user define package and access the Member variable of classes that Contained by Package.

❖Answer:-

```
// UserDefinedClass.java
package mypackage;
public class UserDefinedClass {
   public String message = "Hello from UserDefinedClass";
}
import mypackage.UserDefinedClass;
public class TestUserDefinedPackage {
   public static void main(String[] args) {
      UserDefinedClass udc = new UserDefinedClass();
      System.out.println(udc.message);
   }
}
```

- Output:-

Hello from UserDefinedClass

❖Question:-12

WAP to create a thread that Implement the Runable interface.

❖Answer:-

```
public class RunnableThread implements Runnable {
    public void run() {
        System.out.println("Thread is running");
    }

    public static void main(String[] args) {
        RunnableThread rt = new RunnableThread();
        Thread t = new Thread(rt);
        t.start();
    }
}
```

- Output:-

Thread is running.

❖Question:-13

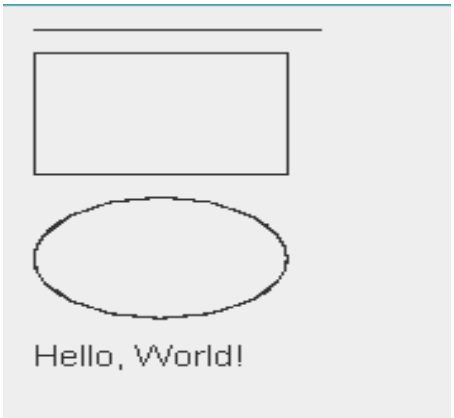
WAP to Draw the line, Rectangle, oval,text using the graphics method.

❖Answer:-

```
import java.awt.*;
import javax.swing.*;

public class DrawShapes extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);

        // Draw line
        g.drawLine(10, 10, 100, 10);

        // Draw rectangle
        g.drawRect(10, 20, 80, 50);

        // Draw oval
        g.drawOval(10, 80, 80, 50);

        // Draw text
```

```
        g.drawString("Hello, World!", 10, 150);}
    public static void main(String[] args) {
        JFrame frame = new JFrame();
        frame.add(new DrawShapes());
        frame.setSize(200, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
    }
}
```

- Output:-



❖Question:-14

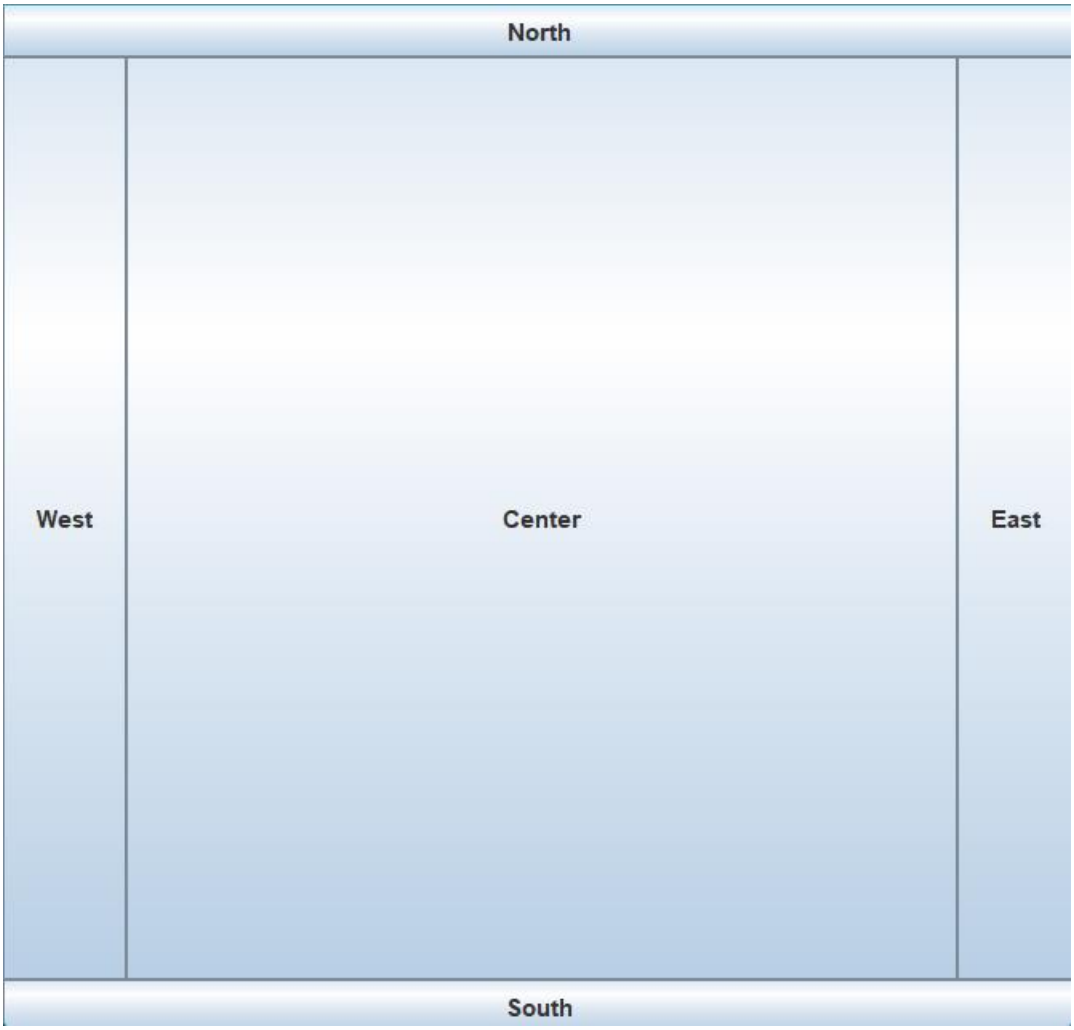
WAP to Implement the flow layout And Border Layout.

❖Answer:-

```
import javax.swing.*;
import java.awt.*;

public class LayoutExample extends JFrame {
    public LayoutExample() {
        setLayout(new FlowLayout());

        add(new JButton("Button 1"));
        add(new JButton("Button 2"));
        add(new JButton("Button 3"));

        setLayout(new BorderLayout());
        add(new JButton("North"), BorderLayout.NORTH);
        add(new JButton("South"), BorderLayout.SOUTH);
        add(new JButton("East"), BorderLayout.EAST);
        add(new JButton("West"), BorderLayout.WEST);
        add(new JButton("Center"), BorderLayout.CENTER);
    }
```

```
    public static void main(String[] args) {
        LayoutExample frame = new LayoutExample();
        frame.setSize(300, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

- Output:-



❖Question:-15

WAP to Implement the GridLayout, CardLayout.

❖Answer:-

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class GridLayoutCardLayoutExample extends JFrame implements
ActionListener {
   CardLayout card;

   public GridLayoutCardLayoutExample() {
      card = new CardLayout();
      setLayout(card);

      JPanel panel1 = new JPanel();
      panel1.setLayout(new GridLayout(2, 2));
      panel1.add(new JButton("1"));
      panel1.add(new JButton("2"));
      panel1.add(new JButton("3"));
```

```
    panel1.add(new JButton("4"));

    JPanel panel2 = new JPanel();
    panel2.add(new JLabel("Card 2"));

    JPanel panel3 = new JPanel();
    panel3.add(new JLabel("Card 3"));

    add(panel1, "Card1");
    add(panel2, "Card2");
    add(panel3, "Card3");

    JButton btn = new JButton("Next");
    btn.addActionListener(this);
    add(btn, BorderLayout.SOUTH);
}

public void actionPerformed(ActionEvent e) {
    card.next(getContentPane());
}

public static void main(String[] args) {
```

```
        GridLayoutCardLayoutExample frame = new
GridLayoutCardLayoutExample();
        frame.setSize(300, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Output:-