1. Write a Java program to print the following triangle of numbers

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

Program

```
class Program1 {
    public static void main(String[] args) {
        int n = 5;
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= i; j++)
                System.out.print(j + " ");
            System.out.println();
        }
    }
}
```

Output

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

2. Write a Java program to list the factorial of the numbers 1 to 10. To calculate the factorial value, use while loop. (Hint Fact of 4 = 4*3*2*1)

Program

```
class Program2 {
   public static void main(String[] args) {
         int fact = 1, i = 1;
         while( i <= 10 )
               System.out.println(i + "! = " + (fact = fact * i++));
   }
}
```

Output

```
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
```

3. Write a Java program

- **To find the area and circumference of the circle by accepting the radius from the user.**
- **To accept a number and find whether the number is Prime or not**

3.1 To find the area and circumference of the circle by accepting the radius from the user.

Program

```
import java.util.Scanner;

public class Program3a {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Circle Radius: ");
        double radius = sc.nextDouble();
        System.out.println("Area of Circle: " + (Math.PI * radius * radius));
        System.out.println("Circumference of Circle: " + (2 * Math.PI * radius));
    }
}
```

Output

```
Enter Circle Radius: 5
Area of Circle: 78.53981633974483
Circumference of Circle: 31.41592653589793
```

3.2 To accept a number and find whether the number is Prime or not

Program

```
import java.util.Scanner;

public class Program3b {
   public static void main(String[] args) {
      System.out.print("Enter a number to check prime or not: ");
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      int flag = 0, m = n/2;
      if(n!=0 && n!=1)
      {
         for (int i = 2; i <= m; i++){
            if(n%i==0)
               flag = 1;
         }
      }
      else
         flag = 1;
      if(flag == 0)
         System.out.println(n + " is a prime number");
      else
         System.out.println(n + " is not a prime number");
   }
}
```

Output 1

```
Enter a number to check prime or not: 5
5 is a prime number
```

Output 2

```
Enter a number to check prime or not: 0
0 is not a prime number
```

Output 3

```
Enter a number to check prime or not: 10
10 is not a prime number
```

4. Write a Java program to demonstrate a division by zero exception

Program

```
public class Program4 {
    public static void main(String[] args) {
        try
        {
            int a = 5, b = 0;
            System.out.println("Quotient: "+ (a/b));
        }
        catch (ArithmeticException e)
        {
            System.out.println("Number cant be divided by zero ");
        }
    }
}
```

Output

```
Number cant be divided by zero
```

5. Write a Java program to implement Inner class and demonstrate its Access protection.

Program

```
class Outer {
   private int outdata = 10;

   void display() {
      Inner in = new Inner();
      System.out.println("Accessing from outer class");
      System.out.println("The value of outdata is " + outdata);
      System.out.println("The value of indata is " + in.indata);
      System.out.println();
   }

   class Inner {
      private int indata = 20;

      void inmethod() {
         System.out.println("Accessing from inner class");
         System.out.println("The sum of indata & outdata is " + (outdata + indata));
      }
   }
}

class Program5 {
   public static void main(String args[]) {
      Outer out = new Outer();
      out.display();
      Outer.Inner in = out.new Inner();
      in.inmethod();
   }
}
```

Output

```
Accessing from outer class
The value of outdata is 10
The value of indata is 20

Accessing from inner class
The sum of indata & outdata is 30
```

6. Write a Java program to demonstrate Constructor Overloading and Method Overloading

Program

```
public class Program6 {
    //Constructor Overloading
    Program6()
    {
        System.out.println("Welcome");
    }

    Program6(String name)
    {
        System.out.println("Welcome "+ name);
    }
    //Method Overloading
    public void add(int a, int b)
    {
        System.out.println("Sum of "+ a +" + "+ b +" = "+ (a+b));
    }

    public void add(double a, double b)
    {
        System.out.println("Sum of "+ a +" + "+ b +" = "+ (a+b));
    }
    public static void main(String[] args) {
        Program6 p1 = new Program6();
        Program6 p2 = new Program6("Yogeesh S");
        p1.add(5, 6);
        p1.add(5.2, 6.4);
    }
}
```

Output

```
Welcome

Welcome Yogeesh S

Sum of 5 + 6 = 11

Sum of 5.2 + 6.4 = 11.600000000000001
```

7. Write a JAVA program to demonstrate Inheritance. Simple Program on Java for the implementation of Multiple inheritance using interfaces to calculate the area of a rectangle and triangle.

Program

```
interface Rectangle{
    void rectangleArea(double w,double h);
}
interface Triangle{
    void triangleArea(double b,double h);
}
class Shapes implements Rectangle,Triangle{
    public void rectangleArea(double w, double h) {
        System.out.println("Rectangle Area is: "+(w*h));
    }
    public void triangleArea(double b, double h) {
        System.out.println("Triangle Area is: "+(0.5*b*h));
    }
}
class Program7 {
    public static void main(String[] args){
        Shapes s = new Shapes();
        s.rectangleArea(5,6);
        s.triangleArea(4,3);
    }
}
```

Output

```
Rectangle Area is: 30.0
Triangle Area is: 6.0
```

8. Write a Java applet program, which handles keyboard event.

Program

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

//<applet code="Program8.java" width=600 height=200></applet>

public class Program8 extends Applet implements KeyListener {
    String msg = "", key;

    public void init() {
        addKeyListener(this);
    }

    public void paint(Graphics g) {
        g.drawString(msg, 150, 100);
    }

    public void keyReleased(KeyEvent ke) {
        showStatus(key + " Key Released");
    }

    public void keyPressed(KeyEvent ke) {
        int keycode = ke.getKeyCode();
        key = ke.getKeyText(keycode);
        repaint();
        showStatus(key + " Key Pressed");
    }

    public void keyTyped(KeyEvent ke) {
        char c = ke.getKeyChar();
        msg += c;
        key = String.valueOf(c);
        repaint();
    }
}
```

To Execution

C:\Users\Yogeesh S\Desktop\Programs>**javac Program8.java**
C:\Users\Yogeesh S\Desktop\Programs>**appletviewer Program8.java**

Output

9. Write a Java Program to create a window when we press

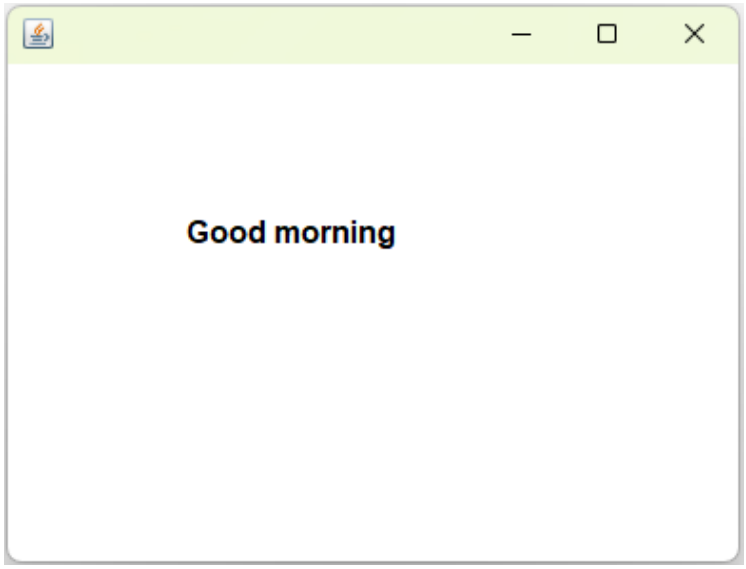
- **M or m the window displays Good Morning**
- **A or a the window displays Good After Noon**
- **E or e the window displays Good Evening**
- **N or n the window displays Good Night**

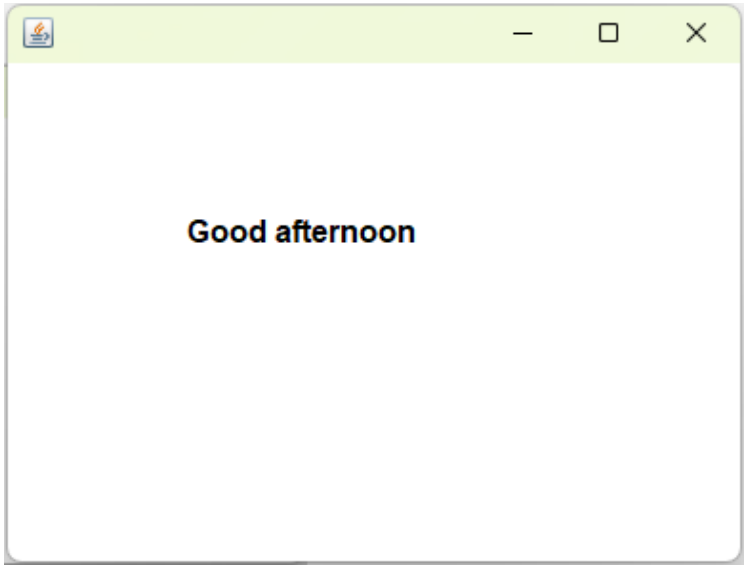
Program

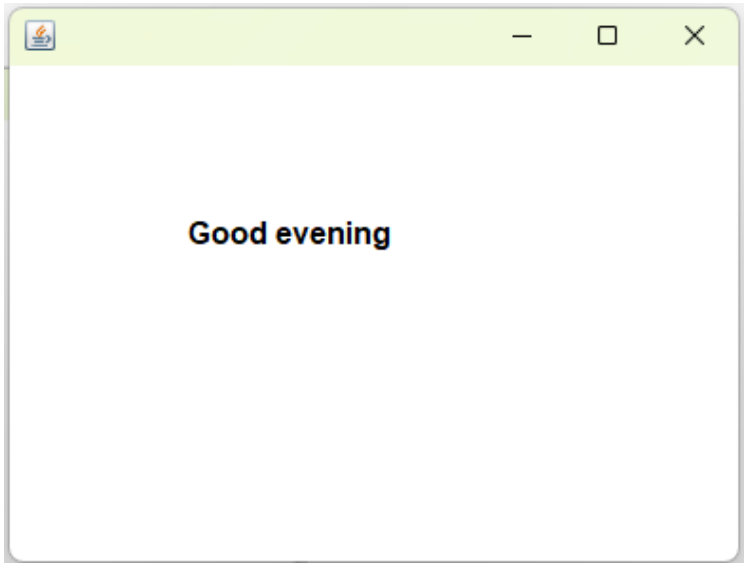
```
import java.awt.*;
import java.awt.event.*;

public class Program9 extends Frame implements KeyListener {
   Label lbl;
   Program9() {
      addKeyListener(this);
      requestFocus();
      lbl = new Label();
      lbl.setBounds(100, 100, 200, 40);
      lbl.setFont(new Font("Calibri", Font.BOLD, 16));
      add(lbl);
      setSize(400, 300);
      setLayout(null);
      setVisible(true);
   }
   public void keyPressed(KeyEvent e) {
      if (e.getKeyChar() == 'M' || e.getKeyChar() == 'm')
         lbl.setText("Good morning");
      else if (e.getKeyChar() == 'A' || e.getKeyChar() == 'a')
         lbl.setText("Good afternoon");
      else if (e.getKeyChar() == 'E' || e.getKeyChar() == 'e')
         lbl.setText("Good evening");
      else if (e.getKeyChar() == 'N' || e.getKeyChar() == 'n')
         lbl.setText("Good night");
   }
   public void keyReleased(KeyEvent e) {
   }
   public void keyTyped(KeyEvent e) {
   }
   public static void main(String[] args) {
      new Program9();
   }
}
```

Output







10. Write a Java program to implement a Queue using user defined Exception Handling (also make use of throw, throws). a. Complete the following: b. Create a package named shape. c. Create some classes in the package representing some common shapes like Square, Triangle, and Circle. d. Import and compile these classes in other program.

Program (Rectangle1.java)

```
package Shape;

public class Rectangle1 {
    private double length, breadth;

    public void setRectangle(double len, double br) {
        length = len;
        breadth = br;
    }

    public void area() {
        double area = length * breadth;
        System.out.println("Area of Rectangle =" + area);
    }
}
```

Program (Square.java)

```
package Shape;

public class Square {
    private double side;

    public void setSquare(double val) {
        side = val;
    }

    public void area() {
        System.out.println("Area of Square=" + (side * side));
    }
}
```

Program(Circle1.java)

```
package Shape;

public class Circle1 {
   private double rad;

   public void setCircle(double radius) {
      rad = radius;
   }

   public void area() {
      double area = (0.5) * 3.14 * rad * rad;
      System.out.println("Area of Rectangle =" + area);
   }
}
```

Program(Program10.java)

```
import Shape.Rectangle1;
import Shape.Square;
import Shape.Circle1;

public class Program10 {
   public static void main(String args[]) {
      Rectangle1 rect = new Rectangle1();
      rect.setRectangle(5.6, 6.4);
      rect.area();
      Square sq = new Square();
      sq.setSquare(10.5);
      sq.area();
      Circle1 round = new Circle1();
      round.setCircle(5.6);
      round.area();
   }
}
```

To Execute

C:\Users\Yogeesh S\Desktop\Programs>**javac -d . Rectangle1.java**
C:\Users\Yogeesh S\Desktop\Programs>**javac -d . Circle1.java**
C:\Users\Yogeesh S\Desktop\Programs>**javac -d . Square.java**
C:\Users\Yogeesh S\Desktop\Programs>**javac Program10.java**
C:\Users\Yogeesh S\Desktop\Programs>**java Program10**

Output

Area of Rectangle =35.839999999999996
Area of Square=110.25
Area of Rectangle =49.2352