OOP's Using Java Lab Programs:

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
```

Solution: Using For Loop

```java
public class NumberTriangle {
    public static void main(String[] args) {
        int rows = 5; // You can change this to any number of rows

        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(j + " ");
            }
            System.out.println(); // Move to the next line after each row
        }
    }
}
```

Note: Convert the For Loop into while Loop and You can use any Looping For this Program.

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).**

Solution-1: Using for Loop

```java
import java.util.Scanner;

public class FactorialWithForLoop {
    public static void main(String[] args) {
        // Create a Scanner object for user input
        Scanner scanner = new Scanner(System.in);

        // Ask the user for input
        System.out.print("Enter a number to calculate the factorials up to: ");
        int num = scanner.nextInt();

        // Initialize factorial value
        int fact = 1;

        // Calculate and print factorial using a for loop
        for (int i = 1; i <= num; i++) {
            fact = fact * i; // Progressive factorial calculation
            System.out.println("Factorial of " + i + " is: " + fact);
        }

        // Close the scanner
        scanner.close();
    }
}
```

Solution-2: Using **while Loop** (Convert the for into while)

```java
import java.util.Scanner;

public class FactorialWithWhileLoop {
    public static void main(String[] args) {
        // Create a Scanner object for user input
        Scanner scanner = new Scanner(System.in);

        // Ask the user for input
        System.out.print("Enter a number to calculate the factorials up to: ");
        int num = scanner.nextInt();

        // Initialize variables
        int fact = 1;
        int i = 1;

        // Calculate and print factorial using a while loop
        while (i <= num) {
            fact = fact * i; // Progressive factorial calculation
            System.out.println("Factorial of " + i + " is: " + fact);
            i++;
        }

        // Close the scanner
        scanner.close();
    }
}
```

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**.

Solution:

- **Program to Find the Area and Circumference of a Circle**:

```java
import java.util.Scanner;

public class CircleProperties {
    public static void main(String[] args) {
        // Create a Scanner object for user input
        Scanner scanner = new Scanner(System.in);

        // Ask the user for the radius
        System.out.print("Enter the radius of the circle: ");
        double radius = scanner.nextDouble();

        // Calculate area and circumference
        double area = Math.PI * radius * radius;
        double circumference = 2 * Math.PI * radius;

        // Display the results
        System.out.println("Area of the circle: " + area);
        System.out.println("Circumference of the circle: " + circumference);

        // Close the scanner
        scanner.close();
    }
}
```

- **Program to Check Whether a Number is Prime**:

```java
import java.util.Scanner;

public class PrimeCheck {
    public static void main(String[] args) {
        // Create a Scanner object for user input
        Scanner scanner = new Scanner(System.in);

        // Ask the user for a number
        System.out.print("Enter a number to check if it is prime: ");
        int num = scanner.nextInt();

        // Check if the number is prime
        boolean isPrime = true;

        if (num <= 1) {
            isPrime = false; // Numbers less than or equal to 1 are not prime
        } else {
            for (int i = 2; i <= num / 2; i++) {
                if (num % i == 0) {
                    isPrime = false;
                    break; // No need to check further if we found a divisor
                }
            }
        }

        // Output the result
        if (isPrime) {
            System.out.println(num + " is a prime number.");
        } else {
            System.out.println(num + " is not a prime number.");
        }

        // Close the scanner
        scanner.close();
    }
}
```

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

Solution:

```java
public class DivisionByZeroDemo {
    public static void main(String[] args) {
        int numerator = 10;
        int denominator = 0;

        try {
            // Attempt to divide by zero
            int result = numerator / denominator;
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            // Handle the division by zero exception
            System.out.println("Error: Division by zero is not allowed.");
            System.out.println("Exception: " + e);
        }
    }
}
```

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

```
class OuterClass {
    int outdata = 10; // Outer class data member

    // Method in outer class
    void display() {
        InnerClass inobj = new InnerClass(); // Creating an object of the inner class
        System.out.println("Accessing from outer class");
        System.out.println("The value of outdata is " + outdata);
        System.out.println("The value of indata is " + inobj.indata); // Accessing inner class data member
    }

    // Inner class
    class InnerClass {
        int indata = 20; // Inner class data member

        // Method in inner class
        void inmethod() {
            System.out.println("Accessing from inner class");

            // Accessing outer class data member.
            System.out.println("The sum of indata & outdata is " + (outdata + indata));
        }
    }
}

public class InnerClassDemo {

    public static void main(String[] args) {
        // Creating an object of the outer class
        OuterClass outobj = new OuterClass();
        outobj.display(); // Calling display method of outer class

        // Creating an object of the inner class and calling its method
        OuterClass.InnerClass inobj1 = outobj.new InnerClass(); // Using outer object to create an inner object
        inobj1.inmethod(); // Calling the method of the inner class

    }
}
```

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

```java
class Calculator {
    String name;

    Calculator() {
        System.out.println("I am Default Constructor");
    }

    Calculator(String name) {
        this.name = name;
        System.out.println("I am Parameterized Constructor Name: " + name);
    }

    public void add(int a, int b) {
        int res = a + b;

        System.out.println(res);
    }

    public void add(int a, int b, int c) {
        int res = a + b + c;

        System.out.println(res);
    }

}

public class Overloading {

    public static void main(String[] args) {
        // Constructor Overloading In Action.
        Calculator cal1 = new Calculator();
        Calculator cal2 = new Calculator("Pramod");

        // Method Overloading In Action
        cal1.add(2, 3);
        cal1.add(2, 3, 5);

    }
}
```

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

```java
//Interface to calculate area of a rectangle
interface Rectangle {
    void areaOfRectangle(double length, double width);
}

// Interface to calculate area of a triangle
interface Triangle {
    void areaOfTriangle(double base, double height);
}

// Class implementing both interfaces
class Shape implements Rectangle, Triangle {

    // Implementation of Rectangle's area calculation
    @Override
    public void areaOfRectangle(double length, double width) {
        double area = length * width;
        System.out.println("Area of Rectangle: " + area);
    }

    // Implementation of Triangle's area calculation
    @Override
    public void areaOfTriangle(double base, double height) {
        double area = 0.5 * base * height;
        System.out.println("Area of Triangle: " + area);
    }
}

public class MultipleInheritance {

    public static void main(String[] args) {
        // Create object of Shape class
        Shape shape = new Shape();

        // Calculate and print area of rectangle
        shape.areaOfRectangle(5.0, 3.0);

        // Calculate and print area of triangle
        shape.areaOfTriangle(4.0, 6.0);

    }
}
```

8. Write a Java Applet Program, which handles keyboard event

```java
import java.applet.Applet;
import java.awt.Graphics;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;

public class PramodApplet extends Applet implements KeyListener {

    private String message = "";
    private String eventType = "";

    //This method is called when the applet is first loaded
    @Override
    public void init() {
        //This tells the applet that it should listen for key events happening within its window.
        addKeyListener(this); // Add the KeyListener to the applet.  (keyPressed, keyReleased, keyTyped).
        setFocusable(true);   // Ensure the applet can receive keyboard input
    }

    @Override
    public void paint(Graphics g) {
        g.drawString("Press any key...", 30, 30);
        g.drawString(eventType + ": " + message, 20, 50);  // Display key event
    }


    // Handling all 3 KeyEvents (keyPressed, keyReleased, keyTyped)
    //Triggered When: A key is physically pressed down, regardless of whether it produces a printable character.
    @Override
    public void keyPressed(KeyEvent e) {
        eventType = "Key Pressed";
        message = KeyEvent.getKeyText(e.getKeyCode()); // Get the name of the key
        repaint(); // Repaint the applet with the new message
    }

    @Override
    public void keyReleased(KeyEvent e) {
        eventType = "Key Released";
        message = KeyEvent.getKeyText(e.getKeyCode());
        repaint();
    }


//Triggered When: A key produces a character, meaning the key press results in a printable character being typed,
    //like letters, numbers, or symbols.
    @Override
    public void keyTyped(KeyEvent e) {
        eventType = "Key Typed";
        message = String.valueOf(e.getKeyChar()); // Get the character of the key typed
        repaint();
    }

}
```