

## **EXP 1:-**

**package Abc;**

**import java.util.Scanner;**

**public class Factorial {**

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

**System.out.print("Enter the Number: ");**

**Scanner sc=new Scanner(System.in);**

**int no=sc.nextInt();**

**long fact=1;**

**for(int i=1;i<=no;i++){**

**fact=fact\*i;**

**}**

**System.out.println("Factorial of "+no+" is "+fact);**

**}**

**}**

**package Abc;**

**/\*\***

**\***

**\*/**

**public class Prime {**

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

**for(int i = 2; i <= 50; i++) {**

**if(isPrime(i)) {**

**System.out.print(i + " ");**

**}**

**}**

**}**

**public static boolean isPrime(int n) {**

**if (n <= 1) return false;**

**for (int i = 2; i<= n/2; i++) { // Changed loop condition for optimization**

**if (n % i == 0) {**

```

        return false;
    }
}

return true;
}
}

package Abc;


import java.util.Scanner;

/**
 *
 */

public class Sum {

    public static void main(String[] args){

        System.out.print("Enter the Total no : ");

        Scanner sc=new Scanner(System.in);

        int n=sc.nextInt();

        int sum=0;

        float avg=0;

        if(n<0){

            System.out.println("Enter Positive Number");

        }else{

            System.out.println("Enter Numbers: ");

            for(int i=1;i<=n;i++) {

                int a=sc.nextInt();

                sum = sum + a;

            }

        }

        avg=sum/n;

        System.out.println("Sum of "+n+" Numbers is "+sum);

        System.out.println("Average of "+n+" Numbers is "+avg);

    }
}

```

```
}
```

## EXP 2:-

```
package Abc;
import java.util.Scanner;

public class Rectangle {
    private double length;
    private double width;
    private String colour;
    private double area;

    // Constructor
    public Rectangle(double length, double width, String colour) {
        this.length = length;
        this.width = width;
        this.colour = colour;
        this.area = length * width;
    }

    // Getter methods
    public double getLength() {
        return length;
    }

    public double getWidth() {
        return width;
    }

    public String getColour() {
        return colour;
    }

    // Method to find area
    public double findArea() {
        return area;
    }

    // Method to compare two rectangles
    public static String compareRectangles(Rectangle rect1, Rectangle rect2) {
        if (rect1.findArea() == rect2.findArea() &&
rect1.getColour().equals(rect2.getColour())) {
            return "Matching Rectangles";
        } else {
            return "Non-matching Rectangles";
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Accepting input for the first rectangle
        System.out.println("Enter length of first rectangle:");
```

```

    double length1 = scanner.nextDouble();
    System.out.println("Enter width of first rectangle:");
    double width1 = scanner.nextDouble();
    System.out.println("Enter color of first rectangle:");
    String color1 = scanner.next();

    // Accepting input for the second rectangle
    System.out.println("Enter length of second rectangle:");
    double length2 = scanner.nextDouble();
    System.out.println("Enter width of second rectangle:");
    double width2 = scanner.nextDouble();
    System.out.println("Enter color of second rectangle:");
    String color2 = scanner.next();

    scanner.close();

    // Creating Rectangle objects
    Rectangle rect1 = new Rectangle(length1, width1, color1);
    Rectangle rect2 = new Rectangle(length2, width2, color2);

    // Comparing the rectangles
    String result = compareRectangles(rect1, rect2);

    // Displaying the result
    System.out.println(result);
}
}

```

EXP 3:-

```

class Box {

    double width, height, depth;

    // constructor used when all dimensions
    // specified
    Box(double w, double h, double d)
    {
        width = w;
        height = h;
        depth = d;
    }

    // constructor used when no dimensions
    // specified
    Box() { width = height = depth = 0;
    }

    // constructor used when cube is created
    Box(double len) { width = height = depth = len;

```

```

}

// compute and return volume

double volume() {

return width * height * depth;

}

public static void main(String args[])

{

// create boxes using the various

// constructors

Box mybox1 = new Box(10, 20, 15);

Box mybox2 = new Box();

Box mycube = new Box(7);

double vol;

// get volume of first box

vol = mybox1.volume();

System.out.println("Volume of mybox1 is " + vol);

// get volume of second box

vol = mybox2.volume();

System.out.println("Volume of mybox2 is " + vol);

// get volume of cube

vol = mycube.volume();

System.out.println("Volume of mycube is " + vol);

}

}

```

## 2. Method Overloading

```

package first;

// method overloading

public class Sum {

// Overloaded sum(). This sum takes two int parameters

public int sum(int x, int y) {

return (x + y);

}

// Overloaded sum(). This sum takes three int parameters

```

```
public int sum(int x, int y, int z)
{
    return (x + y + z);
}

// Overloaded sum(). This sum takes two double parameters
public double sum(double x, double y)
{
    return (x + y);
}

public static void main(String args[])
{
    Sum s = new Sum();
    System.out.println(s.sum(10, 20));
    System.out.println(s.sum(10, 20, 30));
    System.out.println(s.sum(10.5, 20.5));
}
}
```

**EXP 4:-**

```
class Player {
    String name = "A";
    String email = "abc@gmail.com";
    int age = 20;
    long mno = 9876543210L;
    public void display() {
        System.out.println("This is Player Class");
    }
}

class CricketPlayer extends Player {
    int mp = 35, tl = 600, wkt = 40;
    public void display1() {
        System.out.println("This is Cricket Player Class");
    }
}

class FootballPlayer extends Player {
```

```

int mp = 35, goals = 30;

public void display1() {
    System.out.println("This is Football Player Class");
}
}

class HockeyPlayer extends Player {
    int mp = 35, goals = 44;
    public void display1() {
        System.out.println("This is Hockey Player Class");
    }
}

public class Main {
    public static void main(String[] args) {
        CricketPlayer cp = new CricketPlayer();
        HockeyPlayer hp = new HockeyPlayer();
        FootballPlayer fp = new FootballPlayer();
        System.out.println("Player Name: " + cp.name + "\nPlayer Age: "
            + cp.age + "\nPlayer email-id: " + cp.email + "\nPlayer Mobile NO: " + cp.mno);
        cp.display();
        System.out.print("\nCricet Info\n");
        System.out.println("Matches Played: " + cp.mp + "\nRuns scored: " + cp.tl +
            "\nWickets Taken: " + cp.wkt);
        cp.display1();
        System.out.println("\nFootball Info");
        System.out.println("Matches Played: " + fp.mp + "\nGoals scored: " + fp.goals);
        fp.display1();
        System.out.print("\nHockey Info");
        System.out.println("\nMatches Played: " + hp.mp + "\nGoals scored: " + hp.goals);
        hp.display1();
    }
}

```

**EXP 5:-**

**// geometry/Rectangle.java**

```
package geometry;
```

```
public class Rectangle {  
    private double length;  
    private double width;  
  
    public Rectangle(double length, double width) {  
        this.length = length;  
        this.width = width;  
    }  
  
    public double calculateArea() {  
        return length * width;  
    }  
}
```

```
// Main.java
```

```
import java.util.Scanner;  
  
import geometry.Rectangle;  
  
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.print("Enter the length of the rectangle: ");  
        double length = scanner.nextDouble();  
  
        System.out.print("Enter the width of the rectangle: ");  
        double width = scanner.nextDouble();  
  
        Rectangle rectangle = new Rectangle(length, width);
```



```
double area = rectangle.calculateArea();
```

```
System.out.println("The area of the rectangle is: " + area);
```

```
scanner.close();
```

```
}
```

```
}
```

**EXP 6 :-**

```
interface Manager{
```

```
public void display();
```

```
}
```

```
interface Employee extends Manager{
```

```
public void work();
```

```
}
```

```
class Interface1 implements Employee{
```

```
public void display() {
```

```
System.out.println("Interface 1");
```

```
}
```

```
public void work() {
```

```
System.out.println("Work1");
```

```
}
```

```
}
```

```
class Interface2 implements Employee{
```

```
public void display() {
```

```
System.out.println("Interface 2");
```

```
}
```

```
public void work() {
```

```
System.out.println("Work2");
```

```
}
```

```
}
```

```
public class Main{
```

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

```
Employee x=new Interface1();
```

```
Employee y=new Interface2();
```

```
x.display();
```

```
x.work();
```

```
y.display();
```

```
y.work();
```

```
}
```

```
}
```

```
EXP 7:-
```

```
public class Exception {
```

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

```
try {
```

```
int divisionresult=50/0;
```

```
System.out.println("division result:"+divisionresult);
```

```
}catch(ArithmeticException e) {
```

```
System.out.println("Arithmetic Exception"+e.getMessage());
```

```
}
```

```
try {
```

```
String name=null;
```

```
System.out.println(name.length());
```

```
}catch(NullPointerException e){
```

```
System.out.println("NullPointerException"+e.getMessage());
```

```
}
```

```
try {
```

```
int[] a= {1,2,3};
```

```
System.out.println(a[10]);
```

```
}catch(ArrayIndexOutOfBoundsException e) {
```

```
System.out.println("ArrayIndexOutOfBoundsException"+e.getMessage());
```

```
}
```

```
try {
```

```
int a=Integer.parseInt("number");
```

```
System.out.println(a);
```

```
}catch(NumberFormatException e) {
```

```
System.out.println("NumberFormatException"+e.getMessage());
```

```
}
```

```
try {  
    Object object=new Object();  
    String string=(String) object;  
}catch(ClassCastException e) {  
    System.out.println("ClassCastException"+e.getMessage());  
}  
  
}  
  
}
```

**EXP 8:-**

```
class Counter {  
    private int count = 0;  
    public synchronized void increment() {  
        count++;  
        notify(); // Notify waiting thread after increment  
    }  
    public synchronized int getCount() {  
        return count;  
    }  
}  
  
class IncrementThread extends Thread {  
    private Counter counter;  
    public IncrementThread(Counter counter) {  
        this.counter = counter;  
    }  
    public void run() {  
        for (int i = 0; i < 5; i++) {  
            counter.increment();  
            try {  
                Thread.sleep(3000);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```

    }
}

class PrintThread extends Thread {
    private Counter counter;

    public PrintThread(Counter counter) {
        this.counter = counter;
    }

    public void run() {
        for (int i = 0; i < 5; i++) {
            synchronized (counter) {
                try {
                    counter.wait(); // Wait for notification from IncrementThread
                    System.out.println("Counter value: " + counter.getCount());
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}

public class ThreadCommunicationExample {
    public static void main(String[] args) {
        Counter counter = new Counter();
        IncrementThread incrementThread = new IncrementThread(counter);
        PrintThread printThread = new PrintThread(counter);
        incrementThread.start();
        printThread.start();
    }
}

```