Factorial Program

```
public class FactorialExample {
    public static void main(String[] args) {
        int i, fact = 1;
        int number = 5; // It is the number to calculate factorial
        for (i = 1; i <= number; i++) {
            fact = fact * i;
        }
        System.out.println("Factorial of " + number + " is: " + fact);
    }
}
```

Output: Factorial of 5 is: 120

Program for prime Numbers

```
public class PrimeNumbers {
    public static void main(String[] args) {
        int i, j, flag;
        int N = 50;
        System.out.println("Prime numbers between 1 and " + N + " are:");
        for (i = 1; i <= N; i++) {
            if (i == 1 || i == 0)
                continue;
            flag = 1;
            for (j = 2; j <= i / 2; ++j) {
                if (i % j == 0) {
                    flag = 0;
                    break;
                }
            }
            if (flag == 1)
                System.out.print(i + " ");
        }
    }
}
```

Output:

```
Prime numbers between 1 and 50 are:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
```

Program to build calculator

```
import java.util.Scanner;


public class Calculator {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int a, b, c;
        int fact = 1;
        int choice;
        char ans;

        System.out.println("\n Program for Calculator");
        System.out.println("\n Enter the first number");
        a = input.nextInt();
        System.out.println("\n Enter the second number");
        b = input.nextInt();

        do {
            System.out.println("\n\t Main Menu");
            System.out.println("\n 1. Addition \n2. Subtraction \n3.
Multiplication \n4. Division \n5. Factorial");
            System.out.println("\n Enter your choice");
            choice = input.nextInt();

            switch (choice) {
                case 1:
                    c = a + b;
                    System.out.println("\n The Addition is " + c);
                    break;
                case 2:
                    c = a - b;
                    System.out.println("\n The Subtraction is " + c);
                    break;
                case 3:
                    c = a * b;
                    System.out.println("\n The Multiplication is " + c);
                    break;
                case 4:
                    if (b != 0) {
                        c = a / b;
                        System.out.println("\n The Division is " + c);
                    } else {
                        System.out.println("\n Cannot divide by zero");
                    }
                    break;
                case 5:
                    System.out.println("\n Enter the number for computing factorial:");
                    int num = input.nextInt();
                    fact = 1;
                    for (int i = 1; i <= num; i++) {
                        fact = fact * i;
                    }
                    System.out.println("Factorial of given number is: " + fact);
                    break;
                default:
                    System.out.println("\n Invalid choice");
            }

            System.out.println("\n Do you want to continue? (y/n)");
            ans = input.next().charAt(0);
        } while (ans == 'y');
    }
```

```
}
```

Output:

```
Program for Calculator

 Enter the first number
10

 Enter the second number
20

         Main Menu

 1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Factorial

 Enter your choice
1

 The Addition is 30

 Do you want to continue? (y/n)
N
```

Write a program in Java with class rectangle and compare

```
import java.io.*;
import java.util.Scanner;

class Rectangle {
    private double width, length, area;
    private String colour;
    Scanner input = new Scanner(System.in);

    void get_width() throws IOException {
        System.out.print("Enter the width: ");
        width = input.nextDouble();
    }

    void get_length() throws IOException {
        System.out.print("Enter the length: ");
        length = input.nextDouble();
    }

    void get_colour() throws IOException {
        System.out.print("\nEnter the colour: ");
        colour = input.next();
    }

    double find_area() {
        area = length * width;
        return area;
    }
    String show_colour() {
        return colour;
    }
}

public class RectMatchDemo {
    public static void main(String[] args) throws IOException {
        Rectangle R1, R2;
        R1 = new Rectangle();
        R2 = new Rectangle();

        System.out.println("\nEnter data about first rectangle:");
        R1.get_length();
        R1.get_width();
        R1.get_colour();

        System.out.println("\nEnter data about second rectangle:");
        R2.get_length();
        R2.get_width();
        R2.get_colour();

        String c1 = R1.show_colour();
        String c2 = R2.show_colour();

        if ((R1.find_area() == R2.find_area()) && (c1.compareTo(c2) == 0))
            System.out.println("\n\tMatching Rectangles...");
        else
            System.out.println("\n\tNon-Matching Rectangles...");
    }
}
```

Output:

```
Enter data about first rectangle:
Enter the length: 10
Enter the width: 10

Enter the colour: orange

Enter data about second rectangle:
Enter the length: 55
Enter the width: 10

Enter the colour: blue

        Non-Matching Rectangles...
```

```
    }
```

Method Overloading

```
public class OverloadingDemo {
    public static void main(String[] args) {
        System.out.println("Sum of two integers:");
        Sum(10, 20);
        System.out.println("Sum of two double integers:");
        Sum(10.5, 20.4);
        System.out.println("Sum of three integers:");
        Sum(10, 20, 30);
    }

    public static void Sum(int num1, int num2) {
        int ans;
        ans = num1 + num2;
        System.out.println(ans);
    }

    public static void Sum(double num1, double num2) {
        double ans;
        ans = num1 + num2;
        System.out.println(ans);
    }

    public static void Sum(int num1, int num2, int num3) {
        int ans;
        ans = num1 + num2 + num3;
        System.out.println(ans);
    }
}
```

Output:

```
Sum of two integers:
30
Sum of two double integers:
30.9
Sum of three integers:
60
```

Constructor Overloading

```
public class OverloadingDemo {
    double num1, num2;

    OverloadingDemo() {
        num1 = 0;
        num2 = 0;
    }

    OverloadingDemo(double num1, double num2) {
        this.num1 = num1;
        this.num2 = num2;
    }

    OverloadingDemo(double num1, double num2, double num3) {
        this.num1 = num1;
        this.num2 = num2 + num3;
    }

    public void Sum() {
        double ans;
        System.out.println("num1=" + num1);
        System.out.println("num2=" + num2);
        ans = num1 + num2;
        System.out.println("Sum=" + ans);
        System.out.println();
    }

    public static void main(String[] args) {
        OverloadingDemo obj1 = new OverloadingDemo();
        OverloadingDemo obj2 = new OverloadingDemo(10, 20);
        OverloadingDemo obj3 = new OverloadingDemo(10, 20, 30);
        obj1.Sum();
        obj2.Sum();
        obj3.Sum();
    }
}
```

Output:

```
num1=0.0
num2=0.0
Sum=0.0

num1=10.0
num2=20.0
Sum=30.0

num1=10.0
num2=50.0
Sum=60.0
```

Program for sorting list of Integers

```
import java.util.Scanner;

class SortingDemo {
    public static void main(String[] args) {
        int n, i, j, temp;
        Scanner input = new Scanner(System.in);
        System.out.println("How many numbers are to be sorted?");
        n = input.nextInt();
        int array[] = new int[n];
        System.out.println("Enter integers...");
        for (i = 0; i < n; i++) {
            array[i] = input.nextInt();
        }
        for (i = 0; i < (n - 1); i++) {
            for (j = 0; j < (n - i - 1); j++) {
                if (array[j] > array[j + 1]) { /* for descending order use
< */
                    temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                }
            }
        }
        System.out.println("Sorted list of integers...");
        for (i = 0; i < n; i++) {
            System.out.print(" " + array[i]);
        }
    }
}
```

Output:

```
How many numbers are to be sorted?
5
Enter integers…
40
10
50
20
30
Sorted list of integers…
10 20 30 40 50
```

Program for sorting list of Names

```
import java.util.Scanner;

class SortingDemo1 {
    public static void main(String[] args) {
        int n, i, j;
        Scanner input = new Scanner(System.in);
        System.out.println("How many strings are to be sorted?");
        n = input.nextInt();
        String names[] = new String[n];
        System.out.println("Enter names...");
        for (i = 0; i < n; i++) {
            names[i] = input.next();
        }
        for (i = 0; i < (n - 1); i++) {
            for (j = 0; j < (n - i - 1); j++) {
                if (names[j].trim().compareTo(names[j + 1].trim()) > 0) {
                    String temp = names[j];
                    names[j] = names[j + 1];
                    names[j + 1] = temp;
                }
            }
        }
        System.out.println("Sorted list of names...");
        for (i = 0; i < n; i++) {
            System.out.println(" " + names[i]);
        }
    }
}
```

Output:

```
How many strings are to be sorted?
5
Enter names…
BBB
DDD
CCC
AAA
EEE
Sorted list of names….
AAA BBB CCC DDD EEE
```

Program for addition of two matrices

```java
import java.util.Scanner;
class MatrixDemo {
    public static void main(String[] args) {
        int n;
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the order of matrix");
        n = input.nextInt();
        int a[][] = new int[n][n];
        int b[][] = new int[n][n];
        int c[][] = new int[n][n];

        System.out.println("Enter Matrix a[][]");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                a[i][j] = input.nextInt();
            }
        }

        System.out.println("Enter Matrix b[][]");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                b[i][j] = input.nextInt();
            }
        }

        System.out.println("Performing matrix addition...");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                c[i][j] = a[i][j] + b[i][j];
            }
        }

        System.out.println("\n\n Matrix c[][]...");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                System.out.print(" " + c[i][j]);
            }
            System.out.println("\n");
        }
    }
}
```

Output:

```
Enter the order of matrix
3
Enter Matrix a[][]
1 2 3
4 5 6
7 8 9
Enter Matrix b[][]
1 1 1
1 1 1
1 1 1
Performing matrix addition…
Matrix c[][]…
2 3 4
5 6 7
8 9 10
```

Program to create a player class with inheritance.

```
class Player {
    String name;

    Player(String n) {
        name = n;
    }

    void show() {
        System.out.println("Name: " + name);
    }
}

class Cricket_Player extends Player {
    String role;

    Cricket_Player(String nm, String r) {
        super(nm);
        role = r;
    }

    void display() {
        System.out.println("\n\n\tCricket Player");
        show();
        System.out.println("Role: " + role);
    }
}

class Football_Player extends Player {
    String role;
    Football_Player(String nm, String r) {
        super(nm);
        role = r;
    }

    void display() {
        System.out.println("\n\n\tFootball Player");
        show();
        System.out.println("Role: " + role);
    }
}

class Hockey_Player extends Player {
    String role;

    Hockey_Player(String nm, String r) {
        super(nm);
        role = r;
    }

    void display() {
        System.out.println("\n\n\tHockey Player");
        show();
        System.out.println("Role: " + role);
    }
}

class PlayerDemo {
    public static void main(String[] args) {
        Cricket_Player obj1 = new Cricket_Player("AAA", "Batsman");
        Football_Player obj2 = new Football_Player("BBB", "GoalKeeper");
        Hockey_Player obj3 = new Hockey_Player("CCC", "Captain");
        obj1.display();
        obj2.display();
        obj3.display();
    }
}
```

```
Output:
        Cricket Player
Name: AAA
Role: Batsman


        Football Player
Name: BBB
Role: GoalKeeper


        Hockey Player
Name: CCC
Role: Captain
```

Program to create multiple threads and demonstrate how two threads communicate with each other.

```
class MyClass {
    int n;
    boolean flag = false;

    synchronized int get() {
        if (!flag)
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException!!!");
            }
        System.out.println("Consumer consuming: " + n);
        flag = false;
        notify();
        return n;
    }

    synchronized void put(int n) {
        if (flag)
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException!!!");
            }
        this.n = n;
        flag = true;
        System.out.println("Procedure producing " + n);
        notify();
    }
```

```java
}

class Procedure implements Runnable {
    MyClass obj;

    Procedure(MyClass obj) {
        this.obj = obj;
        new Thread(this, "Procedure").start();
    }

    public void run() {
        for (int i = 0; i < 10; i++) {
            obj.put(i);
        }
    }
}

class Consumer implements Runnable {
    MyClass obj;

    Consumer(MyClass obj) {
        this.obj = obj;
        new Thread(this, "Consumer").start();
    }

    public void run() {
        for (int i = 0; i < 10; i++) {
            obj.get();
        }
    }
}
```

```
class InterThread {
    public static void main(String[] args) {
        MyClass TObj = new MyClass();
        new Procedure(TObj);
        new Consumer(TObj);
    }
}
```

Output:

```
Procedure producing 0
Consumer consuming: 0
Procedure producing 1
Consumer consuming: 1
Procedure producing 2
Consumer consuming: 2
Procedure producing 3
Consumer consuming: 3
Procedure producing 4
Consumer consuming: 4
Procedure producing 5
Consumer consuming: 5
Procedure producing 6
Consumer consuming: 6
Procedure producing 7
Consumer consuming: 7
Procedure producing 8
Consumer consuming: 8
Procedure producing 9
Consumer consuming: 9
```

Program which use try and catch for exception handling.

```
class ExceptionDemo {
    public static void main(String[] args) {
        try {
            int a, b;
            a = 5;
            b = a / 0;
        } catch (ArithmeticException e) {
            System.out.println("Divide by zero\n");
            System.out.println("...Executed catch statement");
        }
    }
}
```

Output:

```
Divide by zero

...Executed catch statement
```

Program to draw oval, rectangle, line, text using graphics class.

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

/*
<applet code="GraphicsDemo" width=300 height=300>
</applet>
*/

public class GraphicsDemo extends Applet {
    public void paint(Graphics g) {
        g.drawLine(70, 30, 200, 30);
        g.drawString("Line", 120, 50);
        g.drawOval(10, 10, 50, 100);
        g.drawString("Oval", 25, 130);
        g.drawRect(70, 70, 70, 70);
        g.drawString("Rectangle", 80, 160);
        g.drawString("It is a Demo of various graphics objects", 40, 250);
    }
}
```

Program in which data is read from one file and should be written in another file line by line.

```
import java.io.*;

public class CopyFile {
    private static void copyDemo(String src, String dst) {
        try {
            File f1 = new File(src);
            File f2 = new File(dst);
            InputStream in = new FileInputStream(f1);
            OutputStream out = new FileOutputStream(f2);
            byte[] buff = new byte[1024];
            int len;
            len = in.read(buff);
            while (len > 0) {
                out.write(buff, 0, len); // writing bytes to the destination file
                len = in.read(buff); // reading the remaining content of the file
            }
            in.close();
            out.close();
            System.out.println("File copied.");
        } catch (FileNotFoundException ex) {
            System.out.println(ex.getMessage() + " in the specified directory.");
            System.exit(0);
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
    public static void main(String[] args) {
        copyDemo(args[0], args[1]);
    }
}
```