Program 1

```
import java.io.*;
class GFG {
    static int Series(int n) {
        int i;
        int sums = 0;
        for (i = 1; i <= n; i++)
            sums += 1 / (i * i); // This will still use integer division
        return sums;
    }
    public static void main(String[] args) {
        int n = 3;
        int res = Series(n);
        System.out.println(res);
    }
}
```

Program 2

```
import java.io.*;
class GFG {
    public int factorial(int i) {
        if (i == 0)
            return 1;
        return i * factorial(i - 1);
    }
    public static void main(String[] args) {
        int n = 4, i, j;
        GFG g = new GFG();
        for (i = 0; i <= n; i++) {
            for (j = 0; j < n - i; j++) {

                System.out.print(" ");
            }
            for (j = 0; j <= i; j++) {
                System.out.print(" " + (g.factorial(i) / (g.factorial(j) * g.factorial(i - j))));
            }
            System.out.println();
        }
    }
}
```

Program 3

```
import java.util.Scanner;
class Exercise31 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Input first number: ");
        double x = in.nextDouble();
        System.out.print("Input second number: ");
        double y = in.nextDouble();
        System.out.print("Input third number: ");
        double z = in.nextDouble();
        if (x < y && y < z) {
            System.out.println("Increasing order");
        }
        else if (x > y && y > z) {
            System.out.println("Decreasing order");
        }
        else {
         System.out.println("Neither increasing nor decreasing order");
      }
      in.close();
   }
}
```

Program 4

```
import java.util.*;
class Complex {
   int real, imaginary;
   Complex() {
   }
   Complex(int tempReal, int tempImaginary) {
      real = tempReal;
      imaginary = tempImaginary;
   }
   Complex addComp(Complex C1, Complex C2) {
      Complex temp = new Complex();
      temp.real = C1.real + C2.real;
      temp.imaginary = C1.imaginary + C2.imaginary;
      return temp;
   }
   Complex subtractComp(Complex C1, Complex C2) {
      Complex temp = new Complex();
      temp.real = C1.real - C2.real;
      temp.imaginary = C1.imaginary - C2.imaginary;
      return temp;
   }

   void printComplexNumber() {
      System.out.println("Complex number: " + real + " + " + imaginary + "i");
   }
}
class GFG {
   public static void main(String[] args) {
      Complex C1 = new Complex(3, 2);
      C1.printComplexNumber();
      Complex C2 = new Complex(9, 5);
      C2.printComplexNumber();
      Complex C3 = new Complex();
      C3 = C3.addComp(C1, C2);
      System.out.print("Sum of ");
      C3.printComplexNumber();
      C3 = C3.subtractComp(C1, C2);
      System.out.print("Difference of ");
      C3.printComplexNumber();
   }
}
```

Program 5

```
public class MyTime {
    private int hour;   // between 0 and 23
    private int minute;// between 0 and 59
    public MyTime(int hour, int minute) {
        setTime(hour, minute);
    }
    public void setTime(int hour, int minute) {
        setHour(hour);
        setMinute(minute);
    }
```

```
public void setHour(int hour) {
    if (hour >= 0 && hour < 24) {
        this.hour = hour;
    } else {
        throw new IllegalArgumentException("Invalid hour!");
    }
}

public void setMinute(int minute) {
    if (minute >= 0 && minute < 60) {
        this.minute = minute;
    } else {
        throw new IllegalArgumentException("Invalid minute!");
    }
}
public int getHour() {
    return hour;
}
public int getMinute() {
    return minute;
}
@Override
public String toString() {
    return String.format("%02d:%02d", hour, minute);
}
public MyTime nextMinute() {
    if (minute == 59) {
        minute = 0;
        nextHour();
    } else {
            minute++;
        }
        return this;
    }
    public MyTime nextHour() {
        if (hour == 23) {
            hour = 0;
        } else {
            hour++;
        }
        return this;
    }
    public static void main(String[] args) {
        MyTime time = new MyTime(23, 59);
        System.out.println("Current time: " + time);
        System.out.println("Next minute: " + time.nextMinute());
        System.out.println("Next hour: " + time.nextHour());
    }
}
```

Program 6

```
import java.util.Scanner;
class Account {
    public String acc_name;
    public double acc_no;
    public int acc_type;
    public double balance;
    public void getData(String name, double no, int type, double bal) {
        acc_name = name;
        acc_no = no;
        acc_type = type;
        balance = bal;
    }
}
class Savings extends Account {
    public void deposit(double amt) {
        balance += amt;
        System.out.println("Balance after deposit: " + balance);
    }
    public void withdraw(double amt) {
        if (amt > balance) {
            System.out.println("Insufficient balance.");
        } else {
            balance -= amt;
            System.out.println("Balance after withdrawal: " + balance);
        }
    }
    public void interest(int time, int no) {
        double rate = 0.06; // Assuming 6% interest rate
        double intr = balance * Math.pow(1 + rate / no, time * no) - balance;
        System.out.println("Interest calculated: " + intr);
        balance += intr;
        System.out.println("The new balance is: " + balance);
    }
}
class Current extends Account {
    public void deposit(double amt) {
        balance += amt;
        System.out.println("Balance after deposit: " + balance);
    }

    public void withdraw(double amt) {
        if (amt > balance) {
            System.out.println("Insufficient balance.");
        } else {
            balance -= amt;
            System.out.println("Balance after withdrawal: " + balance);
            check(balance);
        }
    }
    public void check(double amt) {
        if (amt < 10000) {
            balance -= 500;
            System.out.println("Penalty applied. Insufficient balance: " + balance);
        }
    }
}
class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int temp = 1;
        while (temp == 1) {
            System.out.println("Enter name:");
            String name = sc.next();
            System.out.println("Enter acc_no:");
            double no = sc.nextDouble();
            System.out.println("Enter acc_type\n0 for Savings\n1 for Current:");
            int type = sc.nextInt();
        System.out.println("Enter initial balance:");
        double amt = sc.nextDouble();
        if (type == 0) {
            Savings s = new Savings();
            s.getData(name, no, type, amt);
            System.out.println("\n1. Deposit\n2. Withdraw\n3. Interest");
            int temp3 = sc.nextInt();
            switch (temp3) {
                case 1:
                    System.out.println("Enter Amount:");
                    double amt1 = sc.nextDouble();
                    s.deposit(amt1);
                    break;
                case 2:
                    System.out.println("Enter Amount:");
                    amt1 = sc.nextDouble();
                    s.withdraw(amt1);
                    break;
                case 3:
                    System.out.println("Enter time period:");
                    int tp = sc.nextInt();
                    System.out.println("Enter number of times interest is compounded per year:");
                    int nof = sc.nextInt();
                    s.interest(tp, nof);
                    break;
                default:
                    System.out.println("Invalid option.");
            }

        } else if (type == 1) {
                Current c = new Current();
                c.getData(name, no, type, amt);
                System.out.println("\n1. Deposit\n2. Withdraw");
                int temp3 = sc.nextInt();
                switch (temp3) {
                    case 1:
                        System.out.println("Enter Amount:");
                        double amt1 = sc.nextDouble();
                        c.deposit(amt1);
                        break;
                    case 2:
                        System.out.println("Enter Amount:");
                        amt1 = sc.nextDouble();
                        c.withdraw(amt1);
                        break;
                    default:
                        System.out.println("Invalid option.");
                }
            } else {
                System.out.println("Invalid account type.");
            }
            System.out.println("To continue, enter 1; to exit, enter 0:");
            temp = sc.nextInt();
        }
        sc.close();
    }
}
```