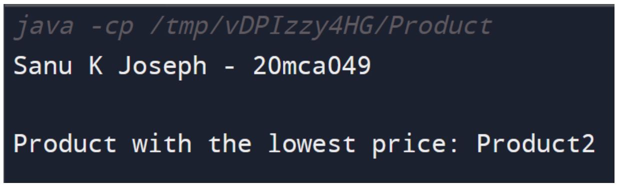
1) Define a class ‘product’ with data members pcode, pname and price. Create 3 objects ofthe class and find the product having the lowest price.

```
public class Product {
        int pcode;
        String pname;
        double price;

        public Product(int pcode, String pname, double price) {
            this.pcode = pcode;
            this.pname = pname;
            this.price = price;
        }

        public static void main(String[] args) {
            Product product1 = new Product(1, "Product1", 10.5);
            Product product2 = new Product(2, "Product2", 5.5);
            Product product3 = new Product(3, "Product3", 8.0);

            Product lowestPriceProduct = findLowestPriceProduct(product1, product2, product3);
            System.out.println("Product with the lowest price: " + lowestPriceProduct.pname);
        }

        public static Product findLowestPriceProduct(Product... products) {
            Product lowestPriceProduct = products[0];
            for (Product product : products) {
                if (product.price < lowestPriceProduct.price) {
                    lowestPriceProduct = product;
                }
            }
            return lowestPriceProduct;
        }
    }
```

Output –

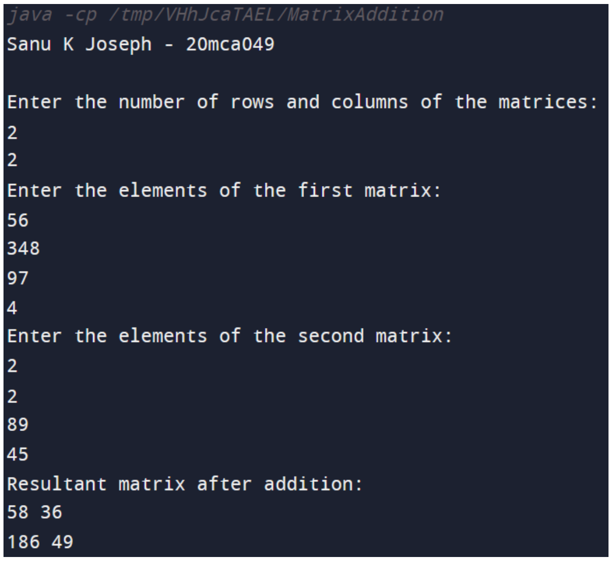


```
java -cp /tmp/vDPIzzy4HG/Product
Sanu K Joseph - 20mca049

Product with the lowest price: Product2
```

2) Read 2 matrices from the console and perform matrix addition.

```
import java.util.Scanner;

    public class MatrixAddition {
        public static int[][] readMatrix(Scanner scanner, int rows, int cols) {
            int[][] matrix = new int[rows][cols];
            for (int i = 0; i < rows; i++) {
                for (int j = 0; j < cols; j++) {
                    matrix[i][j] = scanner.nextInt();
                }
            }
            return matrix;
        }

        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter the number of rows and columns of the matrices:");
            int rows = scanner.nextInt();
            int cols = scanner.nextInt();
            System.out.println("Enter the elements of the first matrix:");
            int[][] matrix1 = readMatrix(scanner, rows, cols);
            System.out.println("Enter the elements of the second matrix:");
            int[][] matrix2 = readMatrix(scanner, rows, cols);
            System.out.println("Resultant matrix after addition:");
            for (int i = 0; i < rows; i++) {
                for (int j = 0; j < cols; j++) {
                    System.out.print(matrix1[i][j] + matrix2[i][j] + " ");
                }
                System.out.println();
            }
            scanner.close();
        }
    }
```

Output –

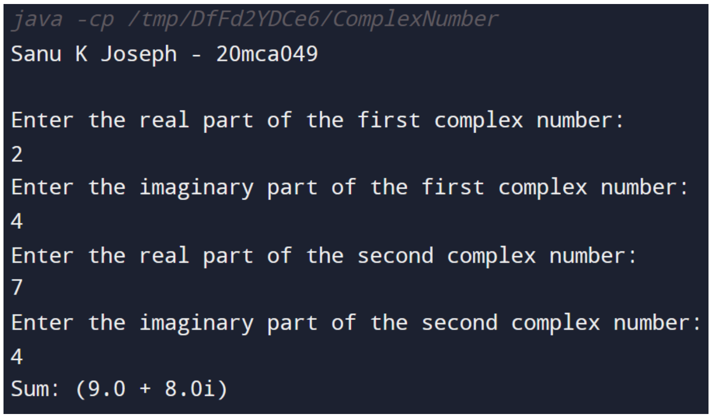


```
java -cp /tmp/VHhJcaTAEL/MatrixAddition
Sanu K Joseph - 20mca049

Enter the number of rows and columns of the matrices:
2
2
Enter the elements of the first matrix:
56
348
97
4
Enter the elements of the second matrix:
2
2
89
45
Resultant matrix after addition:
58 36
186 49
```

3) Add complex numbers

```
import java.util.Scanner;

    public class ComplexNumber {
        private final double real;
        private final double imaginary;

        public ComplexNumber(double real, double imaginary) {
            this.real = real;
            this.imaginary = imaginary;
        }

        public String toString() {
            return "(" + real + " + " + imaginary + "i)";
        }

        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter the real part of the first complex number:");
            double real1 = scanner.nextDouble();
            System.out.println("Enter the imaginary part of the first complex number:");
            double imaginary1 = scanner.nextDouble();
            System.out.println("Enter the real part of the second complex number:");
            double real2 = scanner.nextDouble();
            System.out.println("Enter the imaginary part of the second complex number:");
            double imaginary2 = scanner.nextDouble();
            ComplexNumber num1 = new ComplexNumber(real1, imaginary1);
            ComplexNumber num2 = new ComplexNumber(real2, imaginary2);
            ComplexNumber sum = new ComplexNumber(num1.real + num2.real, num1.imaginary + num2.imaginary);
            System.out.println("Sum: " + sum.toString());
            scanner.close();
        }
    }
```

Output –

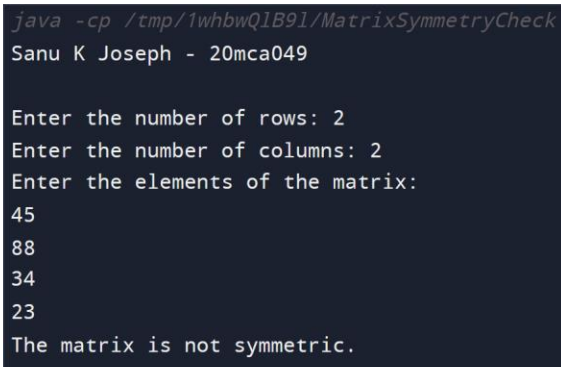


```
java -cp /tmp/DfFd2YDCe6/ComplexNumber
Sanu K Joseph - 20mca049

Enter the real part of the first complex number:
2
Enter the imaginary part of the first complex number:
4
Enter the real part of the second complex number:
7
Enter the imaginary part of the second complex number:
4
Sum: (9.0 + 8.0i)
```

4) Read a matrix from the console and check whether it is symmetric or not.

```
import java.util.Scanner;

    public class MatrixSymmetryCheck {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter the number of rows: ");
            int rows = scanner.nextInt();
            System.out.print("Enter the number of columns: ");
            int cols = scanner.nextInt();
            int[][] matrix = new int[rows][cols];
            System.out.println("Enter the elements of the matrix:");
            for (int i = 0; i < rows; i++) {
                for (int j = 0; j < cols; j++) {
                    matrix[i][j] = scanner.nextInt();
                }
            }
            boolean symmetric = true;
            if (rows != cols) {
                symmetric = false;
            } else {
                for (int i = 0; i < rows; i++) {
                    for (int j = 0; j < i; j++) {
                        if (matrix[i][j] != matrix[j][i]) {
                            symmetric = false;
                            break;
                        }
                    }
                }
            }
            if (symmetric) {
                System.out.println("The matrix is symmetric.");
            } else {
                System.out.println("The matrix is not symmetric.");
            }
            scanner.close();
        }
    }
```

Output –

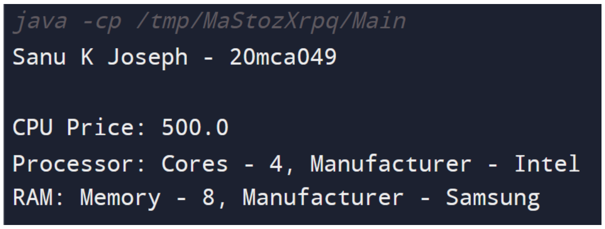


```
java -cp /tmp/1whbwQlB9l/MatrixSymmetryCheck
Sanu K Joseph - 20mca049

Enter the number of rows: 2
Enter the number of columns: 2
Enter the elements of the matrix:
45
88
34
23
The matrix is not symmetric.
```

5) Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

```
class CPU {
    double price;
    static class Processor {
      int cores;
      String manufacturer;
      Processor(int cores, String manufacturer) {
        this.cores = cores;
        this.manufacturer = manufacturer;
      }
      void display() {
        System.out.println("Processor: Cores - " + cores + ", Manufacturer - " + manufacturer);
      }
    }
    static class RAM {
      int memory;
      String manufacturer;
      RAM(int memory, String manufacturer) {
        this.memory = memory;
        this.manufacturer = manufacturer;
      }
      void display() {
        System.out.println("RAM: Memory - " + memory + ", Manufacturer - " + manufacturer);
      }
    }
    CPU(double price) {
      this.price = price;
    }
    void display() {
      System.out.println("CPU Price: " + price);
    }
  }
  public class Main {
    public static void main(String[] args) {
      CPU cpu = new CPU(500);
      CPU.Processor processor = new CPU.Processor(4, "Intel");
      CPU.RAM ram = new CPU.RAM(8, "Samsung");
      cpu.display();
      processor.display();
      ram.display();
    }
  }
```

Output –

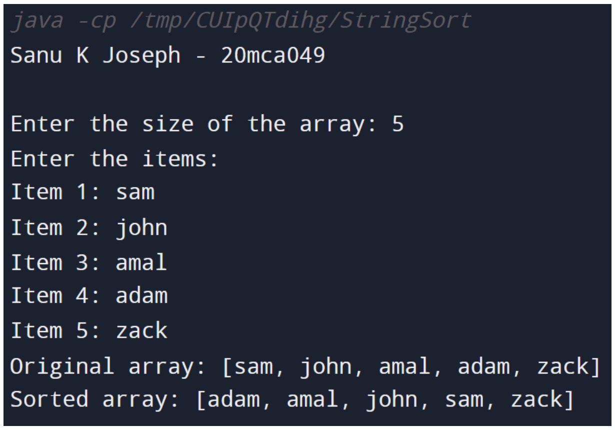


```
java -cp /tmp/MaStozXrpq/Main
Sanu K Joseph - 20mca049

CPU Price: 500.0
Processor: Cores - 4, Manufacturer - Intel
RAM: Memory - 8, Manufacturer - Samsung
```

6) Program to Sort strings.

```
import java.util.Arrays;
    import java.util.Scanner;

    public class StringSort {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter the size of the array: ");
            int size = scanner.nextInt();
            scanner.nextLine();
            String[] strings = new String[size];
            System.out.println("Enter the items:");
            for (int i = 0; i < size; i++) {
                System.out.print("Item " + (i + 1) + ": ");
                strings[i] = scanner.nextLine();
            }
            System.out.println("Original array: " + Arrays.toString(strings));
            Arrays.sort(strings);
            System.out.println("Sorted array: " + Arrays.toString(strings));
            scanner.close();
        }
    }
```

Output –

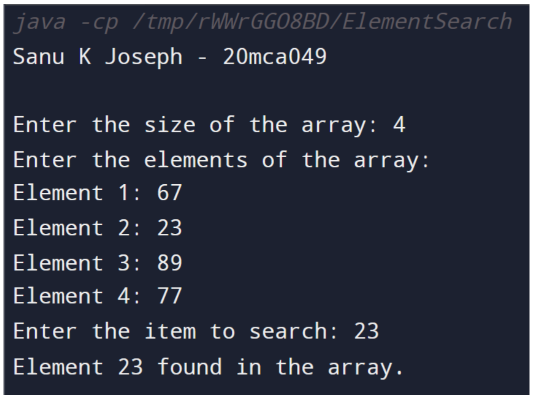


```
java -cp /tmp/CUIpQTdihg/StringSort
Sanu K Joseph - 20mca049

Enter the size of the array: 5
Enter the items:
Item 1: sam
Item 2: john
Item 3: amal
Item 4: adam
Item 5: zack
Original array: [sam, john, amal, adam, zack]
Sorted array: [adam, amal, john, sam, zack]
```

7) Search an element in an array.

```
import java.util.Scanner;

    public class ElementSearch {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter the size of the array: ");
            int size = scanner.nextInt();
            int[] array = new int[size];
            System.out.println("Enter the elements of the array:");
            for (int i = 0; i < size; i++) {
                System.out.print("Element " + (i + 1) + ": ");
                array[i] = scanner.nextInt();
            }
            System.out.print("Enter the item to search: ");
            int target = scanner.nextInt();
            boolean found = false;
            for (int num : array) {
                if (num == target) {
                    found = true;
                    break;
                }
            }
            if (found) {
                System.out.println("Element " + target + " found in the array.");
            } else {
                System.out.println("Element " + target + " not found in the array.");
            }
            scanner.close();
        }
    }
```

Output –

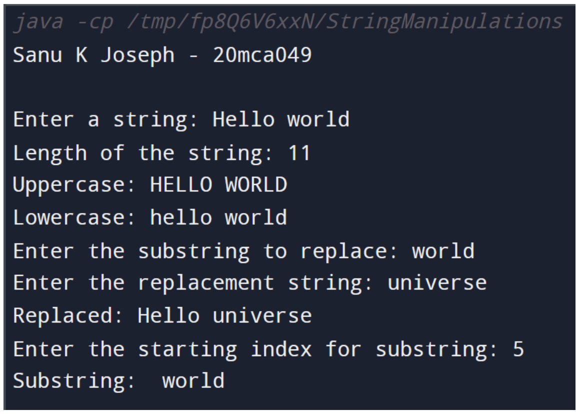


```
java -cp /tmp/rWWrGGO8BD/ElementSearch
Sanu K Joseph - 20mca049

Enter the size of the array: 4
Enter the elements of the array:
Element 1: 67
Element 2: 23
Element 3: 89
Element 4: 77
Enter the item to search: 23
Element 23 found in the array.
```

8) Perform string manipulations.

```
import java.util.Scanner;

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

            System.out.print("Enter a string: ");
            String str = scanner.nextLine();

            int length = str.length();
            System.out.println("Length of the string: " + length);

            String upperCase = str.toUpperCase();
            System.out.println("Uppercase: " + upperCase);

            String lowerCase = str.toLowerCase();
            System.out.println("Lowercase: " + lowerCase);

            System.out.print("Enter the substring to replace: ");
            String substringToReplace = scanner.nextLine();
            System.out.print("Enter the replacement string: ");
            String replacement = scanner.nextLine();
            String replaced = str.replace(substringToReplace, replacement);
            System.out.println("Replaced: " + replaced);

            System.out.print("Enter the starting index for substring: ");
            int startIndex = scanner.nextInt();
            String substring = str.substring(startIndex);
            System.out.println("Substring: " + substring);

            scanner.close();
        }
    }
```

Output –

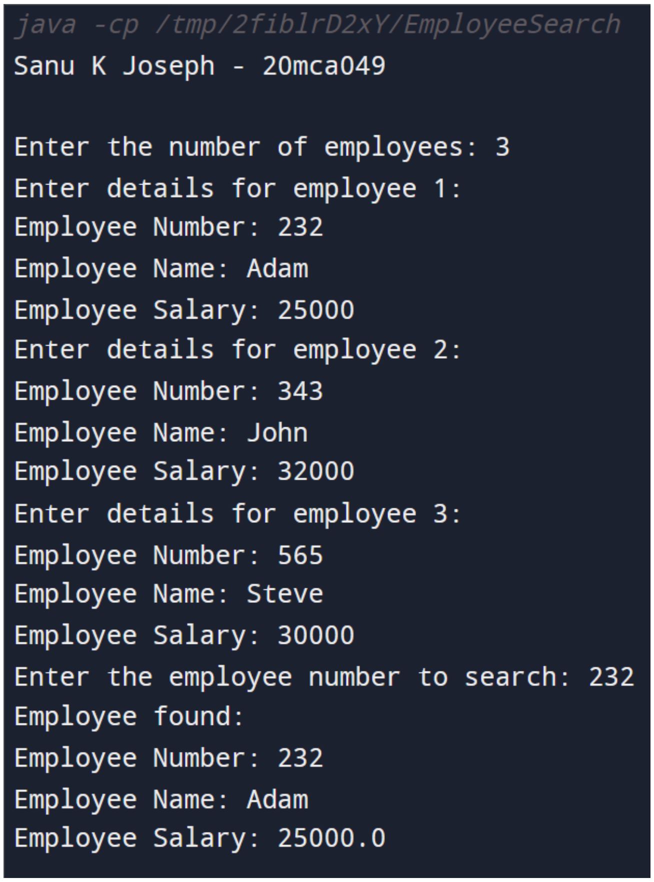


```
java -cp /tmp/fp8Q6V6xxN/StringManipulations
Sanu K Joseph - 20mca049

Enter a string: Hello world
Length of the string: 11
Uppercase: HELLO WORLD
Lowercase: hello world
Enter the substring to replace: world
Enter the replacement string: universe
Replaced: Hello universe
Enter the starting index for substring: 5
Substring:  world
```

9) Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

```
import java.util.Scanner;

  class Employee {
    int eNo;
    String eName;
    double eSalary;

    public Employee(int eNo, String eName, double eSalary) {
      this.eNo = eNo;
      this.eName = eName;
      this.eSalary = eSalary;
    }
  }

  public class EmployeeSearch {
    public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter the number of employees: ");
      int n = scanner.nextInt();
      Employee[] employees = new Employee[n];
      for (int i = 0; i < n; i++) {
        System.out.println("Enter details for employee " + (i + 1) + ":");
        System.out.print("Employee Number: ");
        int eNo = scanner.nextInt();
        scanner.nextLine(); // Consume newline
        System.out.print("Employee Name: ");
        String eName = scanner.nextLine();
        System.out.print("Employee Salary: ");
        double eSalary = scanner.nextDouble();
        employees[i] = new Employee(eNo, eName, eSalary);
      }
      System.out.print("Enter the employee number to search: ");
      int searchNo = scanner.nextInt();
      boolean found = false;
      for (Employee emp : employees) {
        if (emp.eNo == searchNo) {
          found = true;
          System.out.println("Employee found:");
          System.out.println("Employee Number: " + emp.eNo);
          System.out.println("Employee Name: " + emp.eName);
          System.out.println("Employee Salary: " + emp.eSalary);
          break;
        }
      }
      if (!found) {
        System.out.println("Employee with eNo " + searchNo + " not found.");
      }
      scanner.close();
    }
  }
```

Output –

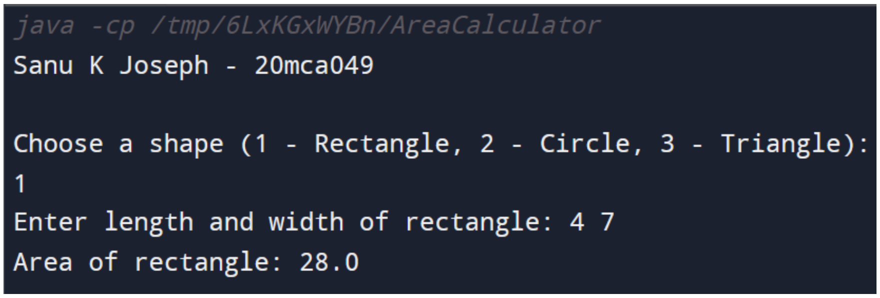


```
java -cp /tmp/2fiblrD2xY/EmployeeSearch
Sanu K Joseph - 20mca049

Enter the number of employees: 3
Enter details for employee 1:
Employee Number: 232
Employee Name: Adam
Employee Salary: 25000
Enter details for employee 2:
Employee Number: 343
Employee Name: John
Employee Salary: 32000
Enter details for employee 3:
Employee Number: 565
Employee Name: Steve
Employee Salary: 30000
Enter the employee number to search: 232
Employee found:
Employee Number: 232
Employee Name: Adam
Employee Salary: 25000.0
```

10) Area of different shapes using overloaded functions

```
import java.util.Scanner;

  public class AreaCalculator {

    public static double calculateArea(double length, double width) {
      return length * width;
    }

    public static double calculateArea(double radius) {
      return Math.PI * radius * radius;
    }

    public static double calculateArea(double base, double height, double factor) {
      return factor * base * height;
    }

    public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.println("Choose a shape (1 - Rectangle, 2 - Circle, 3 - Triangle): ");
      int choice = scanner.nextInt();
      switch (choice) {
        case 1:
          System.out.print("Enter length and width of rectangle: ");
          System.out.println("Area of rectangle: " + calculateArea(scanner.nextDouble(),
scanner.nextDouble()));
          break;
        case 2:
          System.out.print("Enter radius of circle: ");
          System.out.println("Area of circle: " + calculateArea(scanner.nextDouble()));
          break;
        case 3:
          System.out.print("Enter base and height of triangle: ");
          System.out.println("Area of triangle: " + calculateArea(scanner.nextDouble(),
scanner.nextDouble(),0.5));
          break;
        default:
          System.out.println("Invalid choice!");
      }
    }
  }
```

Output –

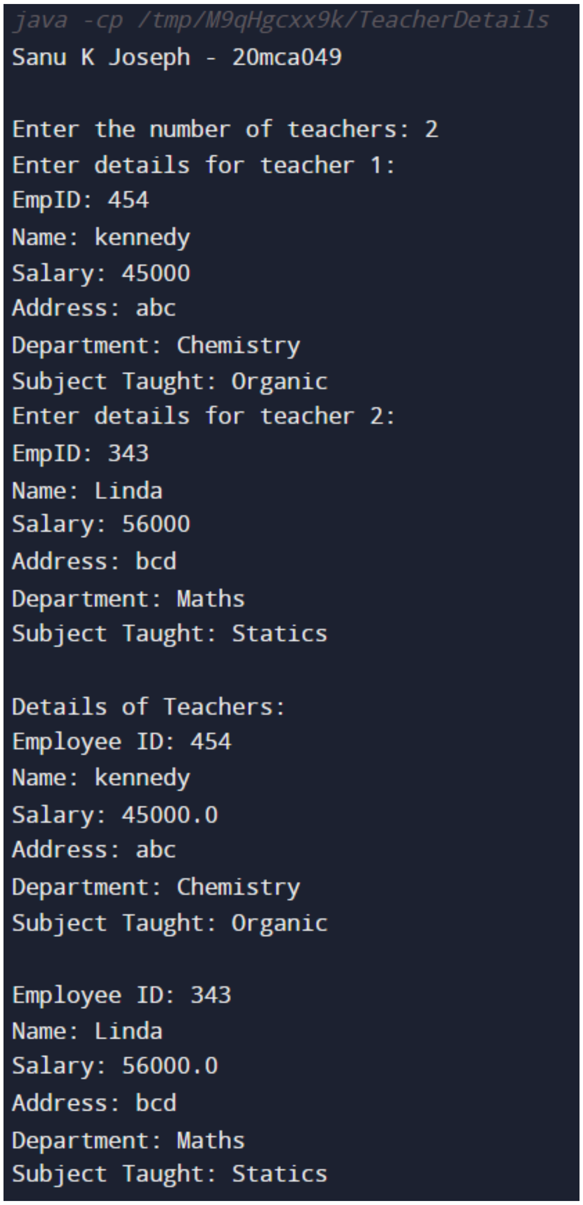


```
java -cp /tmp/6LxKGxWYBn/AreaCalculator
Sanu K Joseph - 20mca049

Choose a shape (1 - Rectangle, 2 - Circle, 3 - Triangle):
1
Enter length and width of rectangle: 4 7
Area of rectangle: 28.0
```

11) Create a class ‘Employee’ with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create anotherclass ‘Teacher’ that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

```
import java.util.Scanner;

    class Employee {
        int empId;
        String name, address;
        double salary;
        Employee(int empId, String name, double salary, String address) {
            this.empId = empId;
            this.name = name;
            this.salary = salary;
            this.address = address;
        }
    }
    class Teacher extends Employee {
        String department;
        String subject;

        Teacher(int empId, String name, double salary, String address, String department, String subject) {
            super(empId, name, salary, address);
            this.department = department;
            this.subject = subject;
        }
        void display() {
            System.out.println("Employee ID: " + empId);
            System.out.println("Name: " + name);
            System.out.println("Salary: " + salary);
            System.out.println("Address: " + address);
            System.out.println("Department: " + department);
            System.out.println("Subject Taught: " + subject);
        }
    }
    public class TeacherDetails {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter the number of teachers: ");
            int numTeachers = scanner.nextInt();
            Teacher[] teachers = new Teacher[numTeachers];
            for (int i = 0; i < numTeachers; i++) {
                System.out.println("Enter details for teacher " + (i + 1) + ":");
                System.out.print("EmpID: ");
                int empId = scanner.nextInt();
                scanner.nextLine();  // Consume newline
                System.out.print("Name: ");
                String name = scanner.nextLine();
                System.out.print("Salary: ");
                double salary = scanner.nextDouble();
                scanner.nextLine();  // Consume newline
                System.out.print("Address: ");
                String address = scanner.nextLine();
                System.out.print("Department: ");
                String department = scanner.nextLine();
                System.out.print("Subject Taught: ");
                String subject = scanner.nextLine();
                teachers[i] = new Teacher(empId, name, salary, address, department, subject);
            }
            System.out.println("\nDetails of Teachers:");
            for (Teacher teacher : teachers) {
                teacher.display();
                System.out.println();
            }
            scanner.close();
        }
    }
```

Output –

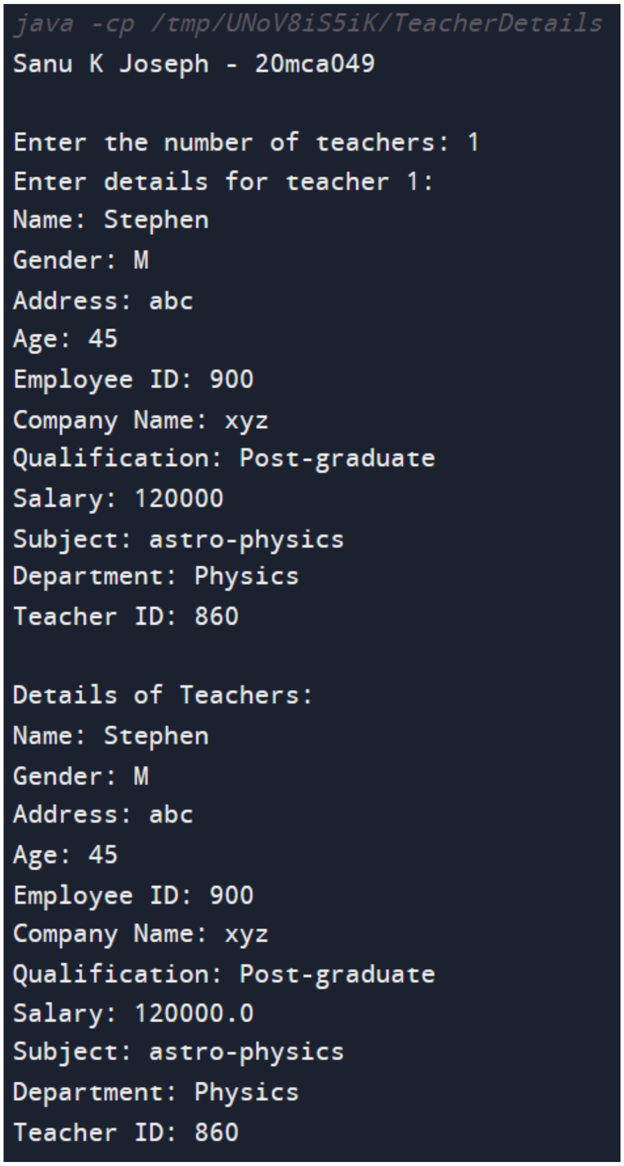


```
java -cp /tmp/M9qHgcxx9k/TeacherDetails
Sanu K Joseph - 20mca049

Enter the number of teachers: 2
Enter details for teacher 1:
EmpID: 454
Name: kennedy
Salary: 45000
Address: abc
Department: Chemistry
Subject Taught: Organic
Enter details for teacher 2:
EmpID: 343
Name: Linda
Salary: 56000
Address: bcd
Department: Maths
Subject Taught: Statics

Details of Teachers:
Employee ID: 454
Name: kennedy
Salary: 45000.0
Address: abc
Department: Chemistry
Subject Taught: Organic

Employee ID: 343
Name: Linda
Salary: 56000.0
Address: bcd
Department: Maths
Subject Taught: Statics
```

12) Create a class 'Person' with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class 'Employee' that inherits the properties of class Person and also contains its own data members like Empid, Company_name, Qualification, Salary and its own constructor. Create another class 'Teacher' that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

```
import java.util.Scanner;

class Person {
    String name, gender, address;
    int age;

    Person(String name, String gender, String address, int age) {
        this.name = name;
        this.gender = gender;
        this.address = address;
        this.age = age;
    }
}

class Employee extends Person {
    int empId;
    String companyName, qualification;
    double salary;

    Employee(String name, String gender, String address, int age, int empId, String companyName, String
qualification, double salary) {
        super(name, gender, address, age);
        this.empId = empId;
        this.companyName = companyName;
        this.qualification = qualification;
        this.salary = salary;
    }
}

class Teacher extends Employee {
    String subject, department;
    int teacherId;

    Teacher(String name, String gender, String address, int age, int empId, String companyName, String
qualification, double salary,
            String subject, String department, int teacherId) {
        super(name, gender, address, age, empId, companyName, qualification, salary);
        this.subject = subject;
        this.department = department;
        this.teacherId = teacherId;
    }

    void display() {
        System.out.println("Name: " + name);
        System.out.println("Gender: " + gender);
        System.out.println("Address: " + address);
        System.out.println("Age: " + age);
        System.out.println("Employee ID: " + empId);
        System.out.println("Company Name: " + companyName);
        System.out.println("Qualification: " + qualification);
        System.out.println("Salary: " + salary);
        System.out.println("Subject: " + subject);
        System.out.println("Department: " + department);
        System.out.println("Teacher ID: " + teacherId);
    }
}

public class TeacherDetails {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of teachers: ");
        int numTeachers = scanner.nextInt();

        Teacher[] teachers = new Teacher[numTeachers];
        for (int i = 0; i < numTeachers; i++) {
            System.out.println("Enter details for teacher " + (i + 1) + ":");
            System.out.print("Name: ");
            String name = scanner.next();
            System.out.print("Gender: ");
            String gender = scanner.next();
            System.out.print("Address: ");
            String address = scanner.next();
            System.out.print("Age: ");
            int age = scanner.nextInt();
            System.out.print("Employee ID: ");
            int empId = scanner.nextInt();
            System.out.print("Company Name: ");
            String companyName = scanner.next();
            System.out.print("Qualification: ");
            String qualification = scanner.next();
            System.out.print("Salary: ");
            double salary = scanner.nextDouble();
            System.out.print("Subject: ");
            String subject = scanner.next();
            System.out.print("Department: ");
            String department = scanner.next();
            System.out.print("Teacher ID: ");
            int teacherId = scanner.nextInt();
            teachers[i] = new Teacher(name, gender, address, age, empId, companyName, qualification, salary, subject,
department, teacherId);
        }

        System.out.println("\nDetails of Teachers:");
        for (Teacher teacher : teachers) {
            teacher.display();
            System.out.println();
        }
        scanner.close();
    }
}
```

Output –

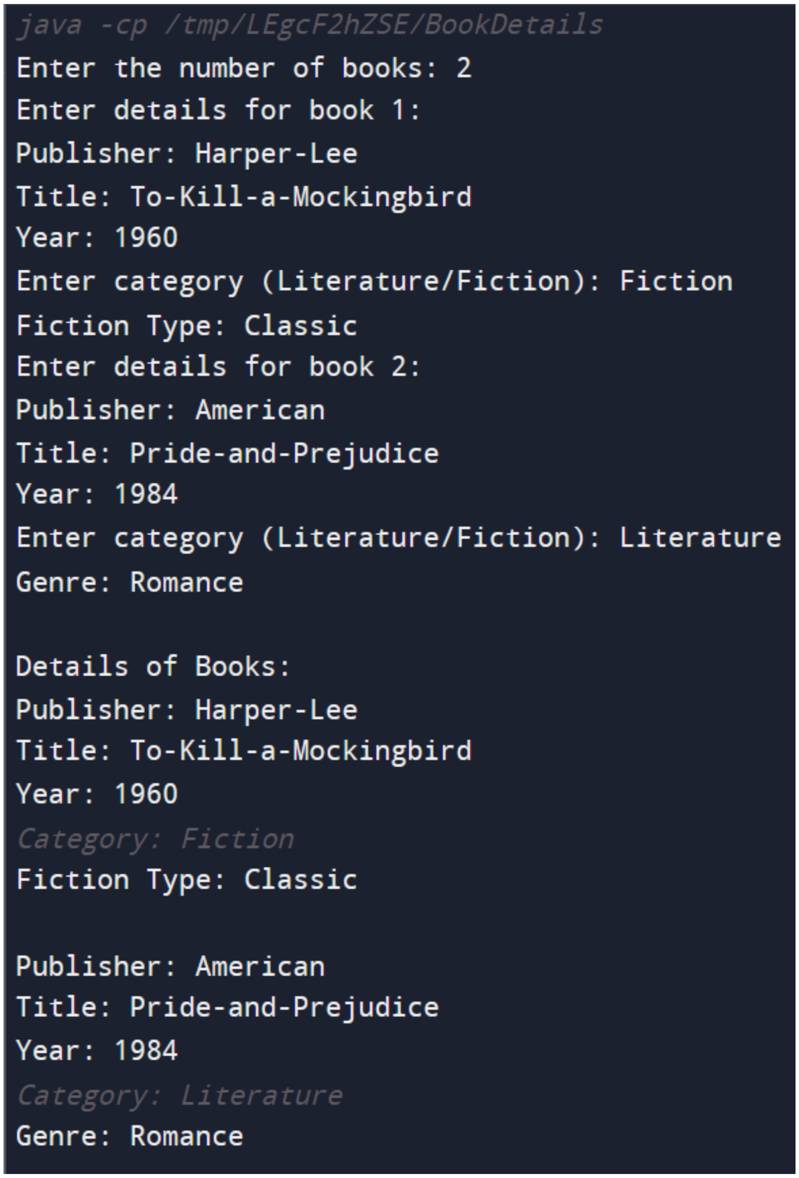


```
java -cp /tmp/UNoV8iS5iK/TeacherDetails
Sanu K Joseph - 20mca049

Enter the number of teachers: 1
Enter details for teacher 1:
Name: Stephen
Gender: M
Address: abc
Age: 45
Employee ID: 900
Company Name: xyz
Qualification: Post-graduate
Salary: 120000
Subject: astro-physics
Department: Physics
Teacher ID: 860

Details of Teachers:
Name: Stephen
Gender: M
Address: abc
Age: 45
Employee ID: 900
Company Name: xyz
Qualification: Post-graduate
Salary: 120000.0
Subject: astro-physics
Department: Physics
Teacher ID: 860
```

13) Write a program has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance.

```
import java.util.Scanner;

class Publisher {
    String publisherName;

    Publisher(String publisherName) {
        this.publisherName = publisherName;
    }
}

class Book extends Publisher {
    String bookTitle;
    int year;

    Book(String publisherName, String bookTitle, int year) {
        super(publisherName);
        this.bookTitle = bookTitle;
        this.year = year;
    }

    void display() {
        System.out.println("Publisher: " + publisherName);
        System.out.println("Title: " + bookTitle);
        System.out.println("Year: " + year);
    }
}

class Literature extends Book {
    String genre;

    Literature(String publisherName, String bookTitle, int year, String genre) {
        super(publisherName, bookTitle, year);
        this.genre = genre;
    }

    void display() {
        super.display();
        System.out.println("Category: Literature");
        System.out.println("Genre: " + genre);
    }
}

class Fiction extends Book {
    String fictionType;

    Fiction(String publisherName, String bookTitle, int year, String fictionType) {
        super(publisherName, bookTitle, year);
        this.fictionType = fictionType;
    }

    void display() {
        super.display();
        System.out.println("Category: Fiction");
        System.out.println("Fiction Type: " + fictionType);
    }
}

public class BookDetails {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        int numBooks = scanner.nextInt();

        Book[] books = new Book[numBooks];
        for (int i = 0; i < numBooks; i++) {
            System.out.println("Enter details for book " + (i + 1) + ":");
            System.out.print("Publisher: ");
            String publisher = scanner.next();
            System.out.print("Title: ");
            String title = scanner.next();
            System.out.print("Year: ");
            int year = scanner.nextInt();
            System.out.print("Enter category (Literature/Fiction): ");
            String category = scanner.next();

            if (category.equalsIgnoreCase("Literature")) {
                System.out.print("Genre: ");
                String genre = scanner.next();
                books[i] = new Literature(publisher, title, year, genre);
            } else if (category.equalsIgnoreCase("Fiction")) {
                System.out.print("Fiction Type: ");
                String fictionType = scanner.next();
                books[i] = new Fiction(publisher, title, year, fictionType);
            } else {
                System.out.println("Invalid category! Skipping book entry.");
                continue;
            }
        }

        System.out.println("\nDetails of Books:");
        for (Book book : books) {
            if (book != null) {
                book.display();
                System.out.println();
            }
        }
        scanner.close();
    }
}
```

Output –

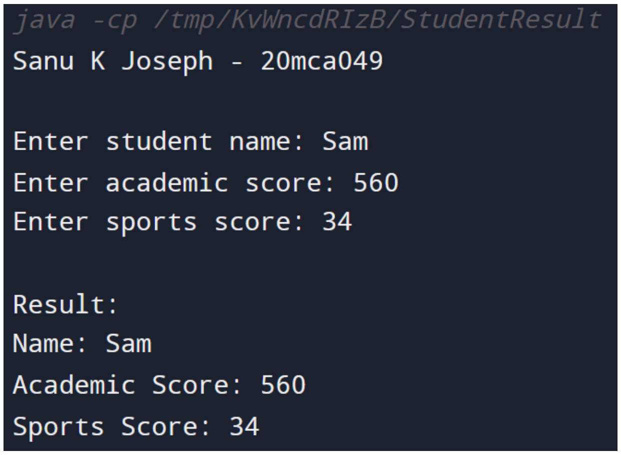


```
java -cp /tmp/LEgcF2hZSE/BookDetails
Enter the number of books: 2
Enter details for book 1:
Publisher: Harper-Lee
Title: To-Kill-a-Mockingbird
Year: 1960
Enter category (Literature/Fiction): Fiction
Fiction Type: Classic
Enter details for book 2:
Publisher: American
Title: Pride-and-Prejudice
Year: 1984
Enter category (Literature/Fiction): Literature
Genre: Romance

Details of Books:
Publisher: Harper-Lee
Title: To-Kill-a-Mockingbird
Year: 1960
Category: Fiction
Fiction Type: Classic

Publisher: American
Title: Pride-and-Prejudice
Year: 1984
Category: Literature
Genre: Romance
```

14) Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

```
import java.util.Scanner;

class Student {
    String name;
    int academicScore;

    Student(String name, int academicScore) {
        this.name = name;
        this.academicScore = academicScore;
    }

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

    int getAcademicScore() {
        return academicScore;
    }
}

class Sports {
    int sportsScore;

    Sports(int sportsScore) {
        this.sportsScore = sportsScore;
    }

    int getSportsScore() {
        return sportsScore;
    }
}

class Result extends Student {
    Sports sports;

    Result(String name, int academicScore, Sports sports) {
        super(name, academicScore);
        this.sports = sports;
    }

    void displayResult() {
        displayStudentInfo();
        System.out.println("Academic Score: " + getAcademicScore());
        System.out.println("Sports Score: " + sports.getSportsScore());
    }
}

public class StudentResult {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter student name: ");
        String name = scanner.nextLine();
        System.out.print("Enter academic score: ");
        int academicScore = scanner.nextInt();
        System.out.print("Enter sports score: ");
        int sportsScore = scanner.nextInt();

        Student student = new Student(name, academicScore);
        Sports sports = new Sports(sportsScore);

        Result result = new Result(name, academicScore, sports);

        System.out.println("\nResult:");
        result.displayResult();

        scanner.close();
    }
}
```

Output –



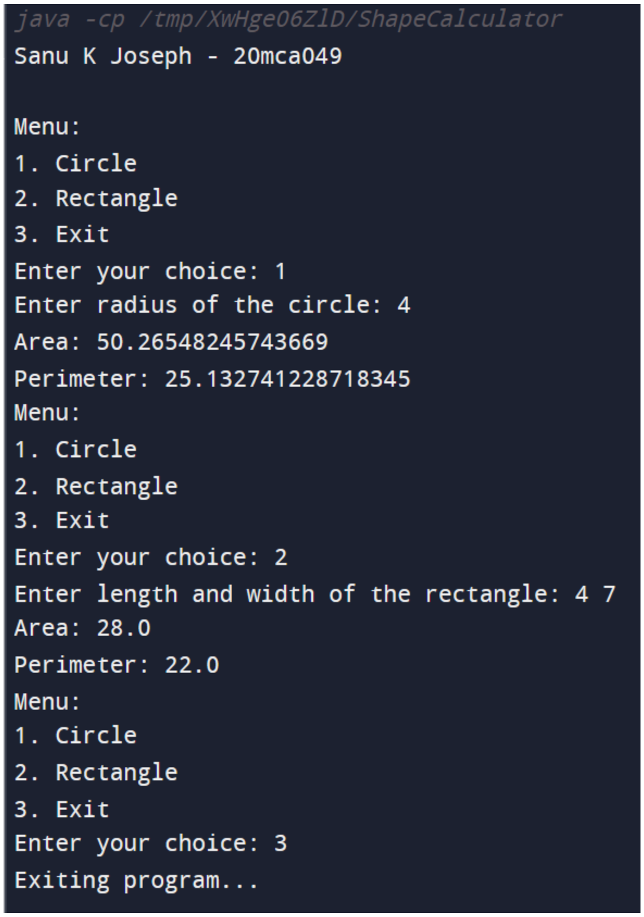
```
java -cp /tmp/KvWncdRIzB/StudentResult
Sanu K Joseph - 20mca049

Enter student name: Sam
Enter academic score: 560
Enter sports score: 34

Result:
Name: Sam
Academic Score: 560
Sports Score: 34
```

15) Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

```
import java.util.Scanner;

interface Shape {
    double area();
    double perimeter();
}

class Circle implements Shape {
    private double radius;
    Circle(double radius) { this.radius = radius; }
    public double area() { return Math.PI * radius * radius; }
    public double perimeter() { return 2 * Math.PI * radius; }
}

class Rectangle implements Shape {
    private double length, width;
    Rectangle(double length, double width) { this.length = length; this.width = width; }
    public double area() { return length * width; }
    public double perimeter() { return 2 * (length + width); }
}

public class ShapeCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;
        do {
            System.out.println("Menu:\n1. Circle\n2. Rectangle\n3. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    System.out.print("Enter radius of the circle: ");
                    Circle circle = new Circle(scanner.nextDouble());
                    System.out.println("Area: " + circle.area());
                    System.out.println("Perimeter: " + circle.perimeter());
                    break;
                case 2:
                    System.out.print("Enter length and width of the rectangle: ");
                    Rectangle rectangle = new Rectangle(scanner.nextDouble(), scanner.nextDouble());
                    System.out.println("Area: " + rectangle.area());
                    System.out.println("Perimeter: " + rectangle.perimeter());
                    break;
                case 3:
                    System.out.println("Exiting program...");
                    break;
                default:
                    System.out.println("Invalid choice!");
            }
        } while (choice != 3);
        scanner.close();
```

```
  }
}
```

Output –



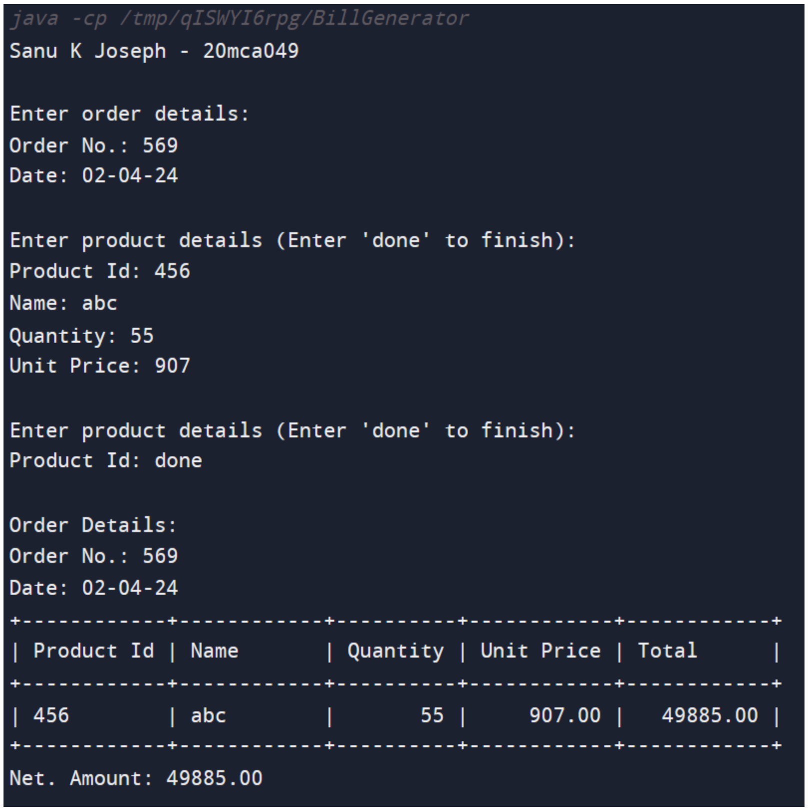
```
java -cp /tmp/XwHge06ZlD/ShapeCalculator
Sanu K Joseph - 20mca049

Menu:
1. Circle
2. Rectangle
3. Exit
Enter your choice: 1
Enter radius of the circle: 4
Area: 50.26548245743669
Perimeter: 25.132741228718345
Menu:
1. Circle
2. Rectangle
3. Exit
Enter your choice: 2
Enter length and width of the rectangle: 4 7
Area: 28.0
Perimeter: 22.0
Menu:
1. Circle
2. Rectangle
3. Exit
Enter your choice: 3
Exiting program...
```

16) Prepare bill with the given format using calculate method from interface.

Product Id	Name	Quantity	Unit Price	Total
101	A	2	25	50
102	B	1	100	100

```
import java.util.*;

class Product {
    private String id, name;
    private int quantity;
    private double unitPrice;

    Product(String id, String name, int quantity, double unitPrice) {
        this.id = id;
        this.name = name;
        this.quantity = quantity;
        this.unitPrice = unitPrice;
    }

    double calculateTotal() {
        return quantity * unitPrice;
    }

    Object[] toTableRow() {
        double total = calculateTotal();
        return new Object[]{id, name, quantity, unitPrice, total};
    }
}

public class BillGenerator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        List<Product> products = new ArrayList<>();
        System.out.println("Enter order details:");
        System.out.print("Order No.: ");
        String orderNo = scanner.nextLine();
        System.out.print("Date: ");
        String date = scanner.nextLine();
        while (true) {
            System.out.println("\nEnter product details (Enter 'done' to finish):");
            System.out.print("Product Id: ");
            String id = scanner.nextLine();
            if (id.equalsIgnoreCase("done")) break;
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Quantity: ");
            int quantity = scanner.nextInt();
            System.out.print("Unit Price: ");
            double unitPrice = scanner.nextDouble();
            scanner.nextLine();  // consume newline
            products.add(new Product(id, name, quantity, unitPrice));
        }
        double total = products.stream().mapToDouble(Product::calculateTotal).sum();
        System.out.println("\nOrder Details:");
        System.out.println("Order No.: " + orderNo);
        System.out.println("Date: " + date);
        printTable(products);
        System.out.printf("Net. Amount: %.2f\n", total);
        scanner.close();
    }

    private static void printTable(List<Product> products) {
        System.out.println("+------------+------------+----------+------------+------------+");
        System.out.println("| Product Id | Name       | Quantity | Unit Price | Total      |");
        System.out.println("+------------+------------+----------+------------+------------+");
        for (Product product : products) {
            Object[] row = product.toTableRow();
            System.out.printf("| %-10s | %-10s | %8d | %10.2f | %10.2f |\n", row);
        }
        System.out.println("+------------+------------+----------+------------+------------+");
    }
}
```

Output –



```
java -cp /tmp/qISWYI6rpg/BillGenerator
Sanu K Joseph - 20mca049

Enter order details:
Order No.: 569
Date: 02-04-24

Enter product details (Enter 'done' to finish):
Product Id: 456
Name: abc
Quantity: 55
Unit Price: 907

Enter product details (Enter 'done' to finish):
Product Id: done

Order Details:
Order No.: 569
Date: 02-04-24
+------------+------------+----------+------------+------------+
| Product Id | Name       | Quantity | Unit Price | Total      |
+------------+------------+----------+------------+------------+
| 456        | abc        |       55 |     907.00 |   49885.00 |
+------------+------------+----------+------------+------------+
Net. Amount: 49885.00
```

17) Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle,Square and Circle. Test the package by finding the area of these figures.

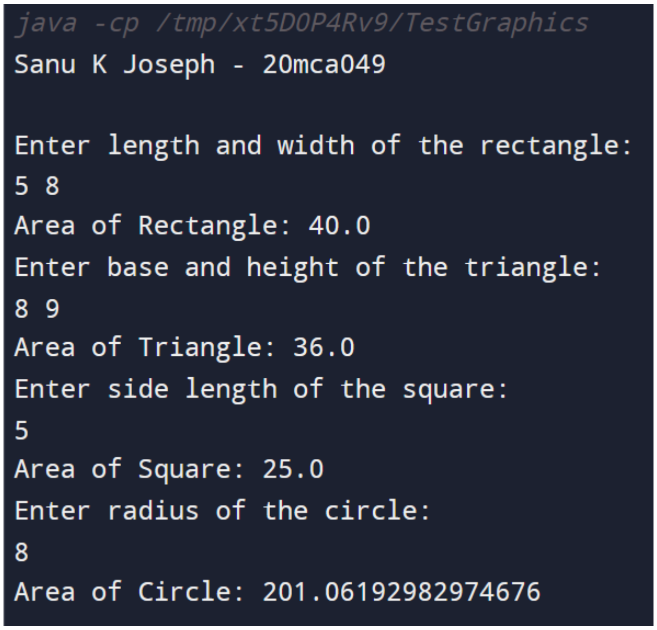
Graphics.java

```
package Graphics;

public interface Shape {
    double area();
}

public class Rectangle implements Shape {
    private double length, width;

    public Rectangle(double length, double width) { this.length = length; this.width = width; }

    public double area() { return length * width; }
}

public class Triangle implements Shape {
    private double base, height;

    public Triangle(double base, double height) { this.base = base; this.height = height; }

    public double area() { return 0.5 * base * height; }
}

public class Square implements Shape {
    private double side;

    public Square(double side) { this.side = side; }

    public double area() { return side * side; }
}

public class Circle implements Shape {
    private double radius;

    public Circle(double radius) { this.radius = radius; }

    public double area() { return Math.PI * radius * radius; }
}
```

17.java

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

public class TestGraphics {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter length and width of the rectangle:");
        double rectangleLength = scanner.nextDouble();
        double rectangleWidth = scanner.nextDouble();
        Rectangle rectangle = new Rectangle(rectangleLength, rectangleWidth);
        System.out.println("Area of Rectangle: " + rectangle.area());
        System.out.println("Enter base and height of the triangle:");
        double triangleBase = scanner.nextDouble();
        double triangleHeight = scanner.nextDouble();
        Triangle triangle = new Triangle(triangleBase, triangleHeight);
        System.out.println("Area of Triangle: " + triangle.area());
        System.out.println("Enter side length of the square:");
        double squareSide = scanner.nextDouble();
        Square square = new Square(squareSide);
        System.out.println("Area of Square: " + square.area());
        System.out.println("Enter radius of the circle:");
        double circleRadius = scanner.nextDouble();
        Circle circle = new Circle(circleRadius);
        System.out.println("Area of Circle: " + circle.area());
        scanner.close();
    }
}
```

Output –



```
java -cp /tmp/xt5DOP4Rv9/TestGraphics
Sanu K Joseph - 20mca049

Enter length and width of the rectangle:
5 8
Area of Rectangle: 40.0
Enter base and height of the triangle:
8 9
Area of Triangle: 36.0
Enter side length of the square:
5
Area of Square: 25.0
Enter radius of the circle:
8
Area of Circle: 201.06192982974676
```

18) Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers

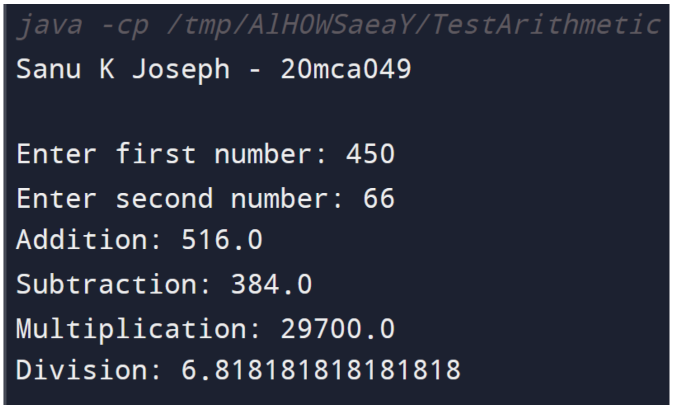
Arithmetic.java

```
package Arithmetic;

public interface Operation {
    double operate(double num1, double num2);
}

public class Addition implements Operation {
    public double operate(double num1, double num2) {
        return num1 + num2;
    }
}

public class Subtraction implements Operation {
    public double operate(double num1, double num2) {
        return num1 - num2;
    }
}

public class Multiplication implements Operation {
    public double operate(double num1, double num2) {
        return num1 * num2;
    }
}

public class Division implements Operation {
    public double operate(double num1, double num2) {
        if (num2 == 0) {
            throw new ArithmeticException("Division by zero is not allowed");
        }
        return num1 / num2;
    }
}
```

18.java

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

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

        System.out.print("Enter first number: ");
        double num1 = scanner.nextDouble();
        System.out.print("Enter second number: ");
        double num2 = scanner.nextDouble();

        System.out.println("Addition: " + new Addition().operate(num1, num2));
        System.out.println("Subtraction: " + new Subtraction().operate(num1, num2));
        System.out.println("Multiplication: " + new Multiplication().operate(num1, num2));
        try {
            System.out.println("Division: " + new Division().operate(num1, num2));
        } catch (ArithmeticException e) {
            System.out.println(e.getMessage());
        }

        scanner.close();
    }
}
```

Output –

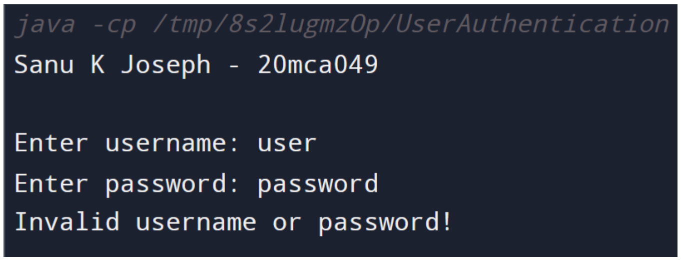


```
java -cp /tmp/AlHOWSaeaY/TestArithmetic
Sanu K Joseph - 20mca049

Enter first number: 450
Enter second number: 66
Addition: 516.0
Subtraction: 384.0
Multiplication: 29700.0
Division: 6.818181818181818
```

19) Write a user defined exception class to authenticate the user name and password

```
import java.util.Scanner;

class AuthenticationException extends Exception {
    AuthenticationException(String message) {
        super(message);
    }
}

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

        String username = "admin";
        String password = "password";

        try {
            System.out.print("Enter username: ");
            String inputUsername = scanner.nextLine();
            System.out.print("Enter password: ");
            String inputPassword = scanner.nextLine();

            if (!inputUsername.equals(username) || !inputPassword.equals(password)) {
                throw new AuthenticationException("Invalid username or password!");
            }
            System.out.println("Authentication successful!");
        } catch (AuthenticationException e) {
            System.out.println(e.getMessage());
        }

        scanner.close();
    }
}
```

Output –

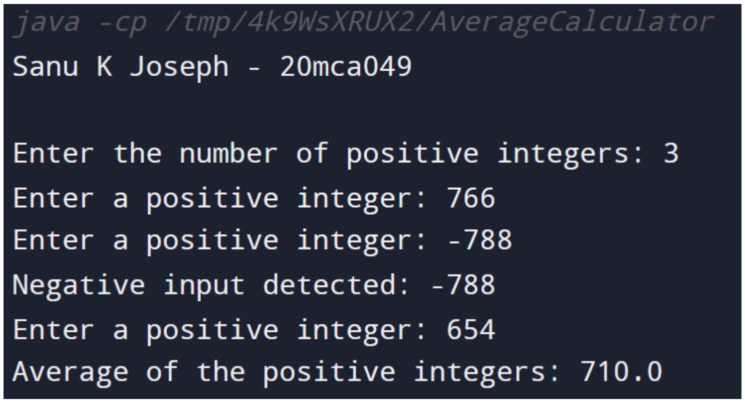


```
java -cp /tmp/8s2lugmzOp/UserAuthentication
Sanu K Joseph - 20mca049

Enter username: user
Enter password: password
Invalid username or password!
```

20) Find the average of N positive integers, raising a user defined exception for each negative input

```
import java.util.Scanner;

class NegativeInputException extends Exception {
    NegativeInputException(String message) {
        super(message);
    }
}

public class AverageCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of positive integers: ");
        int N = scanner.nextInt();
        int sum = 0;
        int count = 0;
        for (int i = 0; i < N; i++) {
            try {
                System.out.print("Enter a positive integer: ");
                int num = scanner.nextInt();
                if (num < 0) {
                    throw new NegativeInputException("Negative input detected: " + num);
                }
                sum += num;
                count++;
            } catch (NegativeInputException e) {
                System.out.println(e.getMessage());
            }
        }
        double average = (count > 0) ? (double) sum / count : 0;
        System.out.println("Average of the positive integers: " + average);
        scanner.close();
    }
}
```

Output –

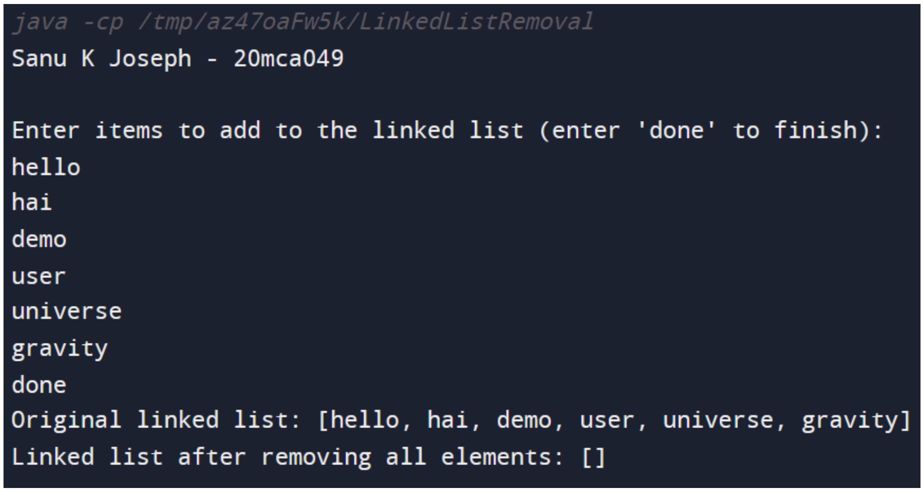


```
java -cp /tmp/4k9WsXRUX2/AverageCalculator
Sanu K Joseph - 20mca049

Enter the number of positive integers: 3
Enter a positive integer: 766
Enter a positive integer: -788
Negative input detected: -788
Enter a positive integer: 654
Average of the positive integers: 710.0
```

21) Program to remove all the elements from a linked list

```
import java.util.LinkedList;
import java.util.Scanner;

public class LinkedListRemoval {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        LinkedList<String> linkedList = new LinkedList<>();

        System.out.println("Enter items to add to the linked list (enter 'done' to finish):");
        while (true) {
            String item = scanner.nextLine();
            if (item.equals("done")) break;
            linkedList.add(item);
        }

        System.out.println("Original linked list: " + linkedList);
        linkedList.clear();
        System.out.println("Linked list after removing all elements: " + linkedList);

        scanner.close();
    }
}
```

Output –

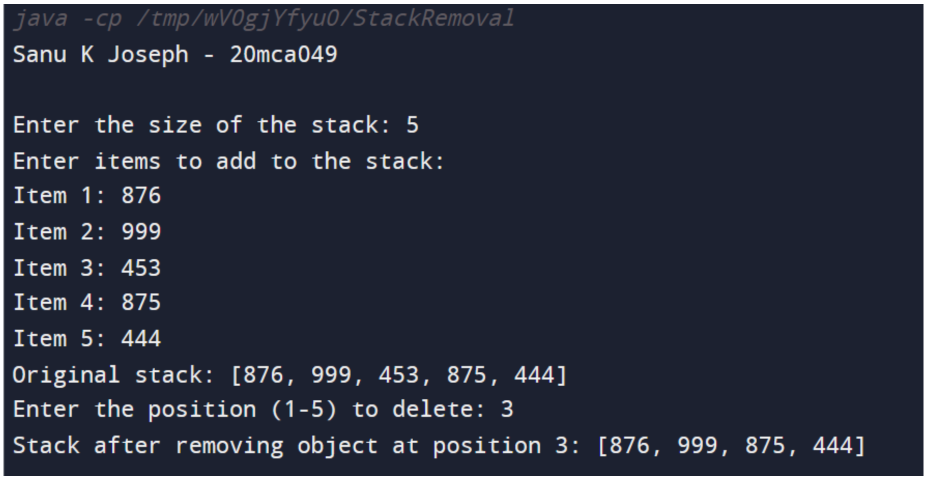


```
java -cp /tmp/az47oaFw5k/LinkedListRemoval
Sanu K Joseph - 20mca049

Enter items to add to the linked list (enter 'done' to finish):
hello
hai
demo
user
universe
gravity
done
Original linked list: [hello, hai, demo, user, universe, gravity]
Linked list after removing all elements: []
```

22) Program to remove an object from the Stack when the position is passed as a parameter

```
import java.util.Scanner;
import java.util.Stack;

public class StackRemoval {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Stack<String> stack = new Stack<>();
        System.out.print("Enter the size of the stack: ");
        int size = scanner.nextInt();
        scanner.nextLine(); // Consume newline character
        System.out.println("Enter items to add to the stack:");
        for (int i = 0; i < size; i++) {
            System.out.print("Item " + (i + 1) + ": ");
            stack.push(scanner.nextLine());
        }
        System.out.println("Original stack: " + stack);
        System.out.print("Enter the position (1-" + stack.size() + ") to delete: ");
        int positionToRemove = scanner.nextInt();
        if (positionToRemove >= 1 && positionToRemove <= stack.size()) {
            Stack<String> tempStack = new Stack<>();
            for (int i = 1; i < positionToRemove; i++) {
                tempStack.push(stack.pop()); // Store elements up to the specified position in tempStack
            }
            stack.pop(); // Remove element at specified position
            while (!tempStack.isEmpty()) {
                stack.push(tempStack.pop()); // Push back elements from tempStack to stack
            }
            System.out.println("Stack after removing object at position " + positionToRemove + ": " + stack);
        } else {
            System.out.println("Invalid position. No object removed.");
        }
        scanner.close();
    }
}
```

Output –

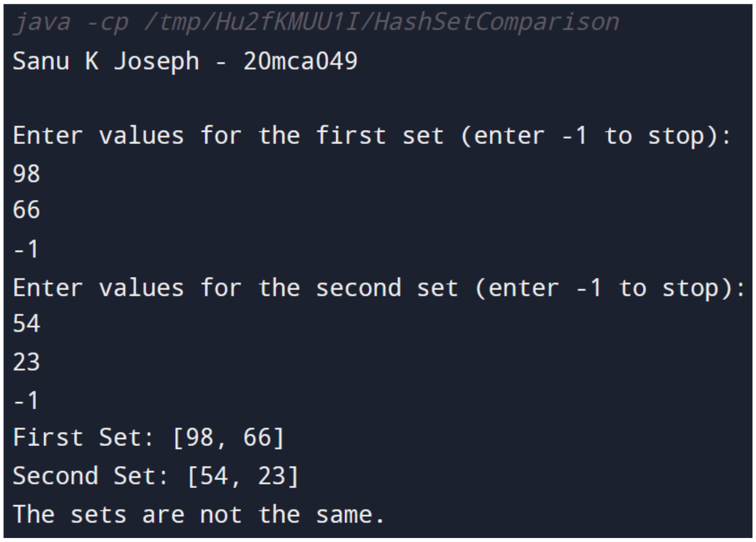


```
java -cp /tmp/wV0gjYfyu0/StackRemoval
Sanu K Joseph - 20mca049

Enter the size of the stack: 5
Enter items to add to the stack:
Item 1: 876
Item 2: 999
Item 3: 453
Item 4: 875
Item 5: 444
Original stack: [876, 999, 453, 875, 444]
Enter the position (1-5) to delete: 3
Stack after removing object at position 3: [876, 999, 875, 444]
```

23) Write a Java program to compare two hash set

```
import java.util.HashSet;
import java.util.Scanner;

public class HashSetComparison {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        HashSet<Integer> set1 = new HashSet<>();
        System.out.println("Enter values for the first set (enter -1 to stop):");
        int value;
        while ((value = scanner.nextInt()) != -1) {
            set1.add(value);
        }
        HashSet<Integer> set2 = new HashSet<>();
        System.out.println("Enter values for the second set (enter -1 to stop):");
        while ((value = scanner.nextInt()) != -1) {
            set2.add(value);
        }
        System.out.println("First Set: " + set1);
        System.out.println("Second Set: " + set2);
        boolean same = set1.equals(set2);
        if (same) {
            System.out.println("The sets are the same.");
        } else {
            System.out.println("The sets are not the same.");
        }
        scanner.close();
    }
}
```

Output –



```
java -cp /tmp/Hu2fKMUU1I/HashSetComparison
Sanu K Joseph - 20mca049

Enter values for the first set (enter -1 to stop):
98
66
-1
Enter values for the second set (enter -1 to stop):
54
23
-1
First Set: [98, 66]
Second Set: [54, 23]
The sets are not the same.
```