**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No:1**

# ----------------------------------------------------------------

**AIM:- Defining simple classes and simple program. Area of Circle, Temp. Convert Celsius to Fahrenheit, patterns.**

**---------------------------------------------------------------------------------------------------------------**

**CODE:-**

class Person {

    String name;

    int age;

    // Constructor to initialize name and age

    public Person(String name, int age) {

        this.name = name;

        this.age = age;

    }

    // Method to display details

    public void display() {

        System.out.println("Name: " + name);

        System.out.println("Age: " + age);

    }

    public static void main(String[] args) {

        // Creating instances of Person

        Person person = new Person("Milind", 22);

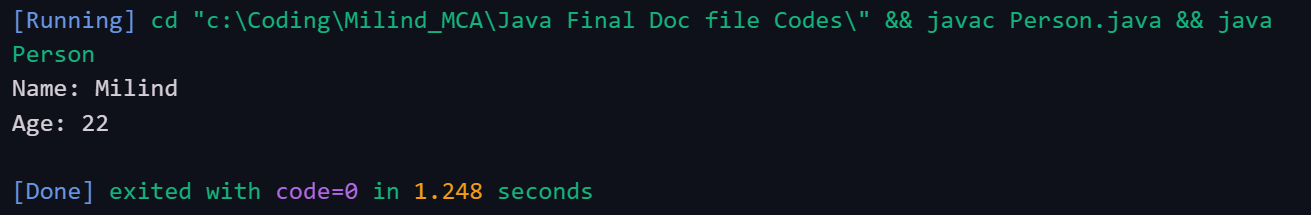
        // Displaying details

        person.display();

    }

}

**Output:-**



* **AreaOfCircle.java**

import java.util.Scanner;

class AreaOfCircle {

    public static void main(String[] args) {

        int n = 22, d = 7;

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter radius: ");

        double r = sc.nextDouble();

        if (r > 0) {

            // Using type casting for accurate result

            double area = (double) n / d \* r \* r;

            System.out.println("Area of circle: " + area);

        } else {

            System.out.println("Invalid input. Radius should be greater than zero.");

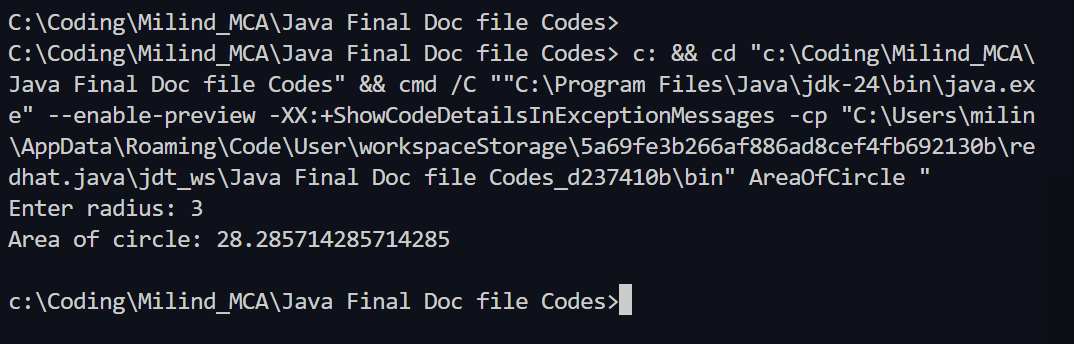
        }

        sc.close();

    }

}

**Output:-**



* **Temp.java**

import java.util.Scanner;

class Temp {

    public static void main(String[] args) {

        float celsius, fahrenheit;

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the Celsius value to convert to Fahrenheit: ");

        celsius = sc.nextFloat();

        fahrenheit = (celsius \* 9 / 5) + 32;

        System.out.println("The converted temperature in Fahrenheit is: " + fahrenheit);

        System.out.println("Enter the Fahrenheit value to convert to Celsius: ");

        fahrenheit = sc.nextFloat();

        celsius = (fahrenheit - 32) \* 5 / 9;

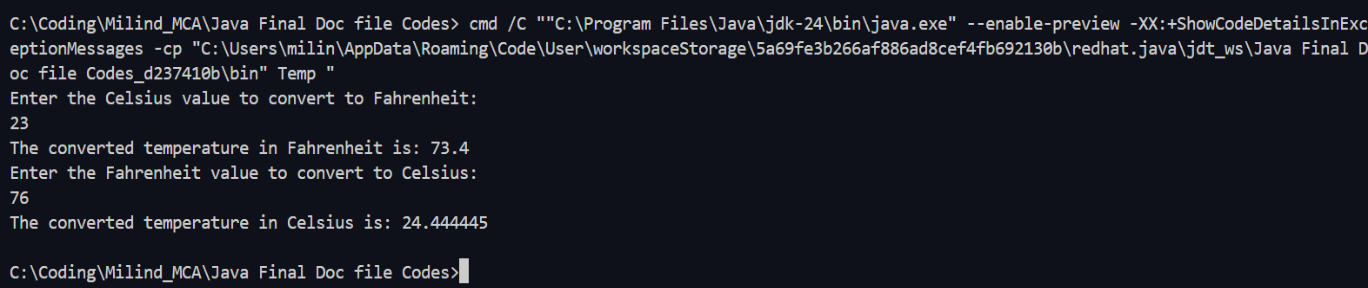
        System.out.println("The converted temperature in Celsius is: " + celsius);

        sc.close();

    }

}

**Output:-**



* **FullPyramid.java**

import java.util.Scanner;

public class FullPyramid {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter rows: ");

        int rows = sc.nextInt();

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

            for (int space = rows - i; space > 0; space--) {

                System.out.print(" ");

            }

            for (int star = 1; star <= (2 \* i - 1); star++) {

                System.out.print("\*");

            }

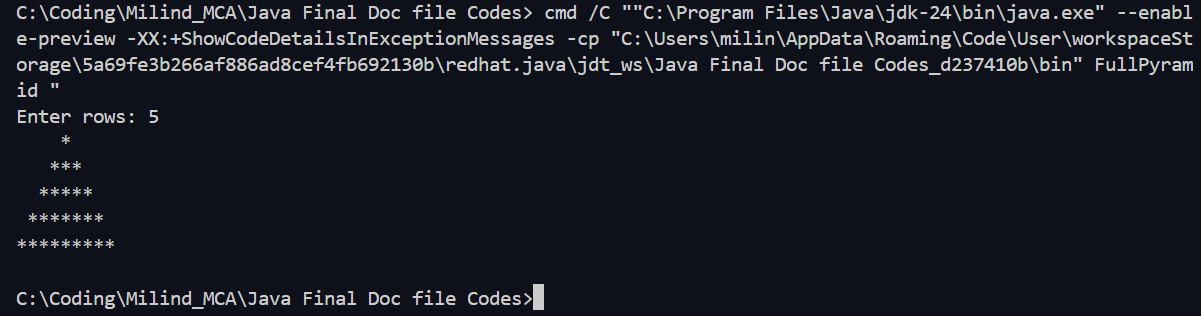
            System.out.println();

        }

    }

}

**Output:-**



* **NumberPattern.java**

import java.util.Scanner;

public class FullPyramid {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter rows: ");

        int rows = sc.nextInt();

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

            for (int space = rows - i; space > 0; space--) {

                System.out.print(" ");

            }

            for (int star = 1; star <= (2 \* i - 1); star++) {

                System.out.print("\*");

            }

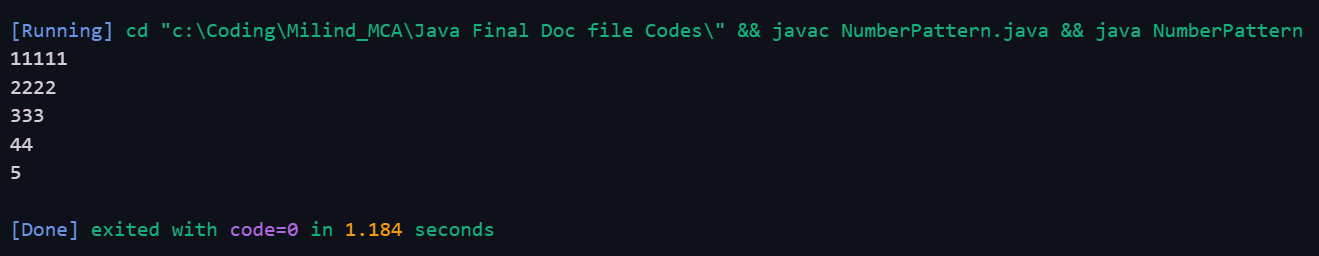
            System.out.println();

        }

    }

}

**Output:-**



* **HalfPyramid.java**

import java.util.Scanner;

public class HalfPyramid {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter rows: ");

        int rows = sc.nextInt();

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

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

                System.out.print("\* ");

            }

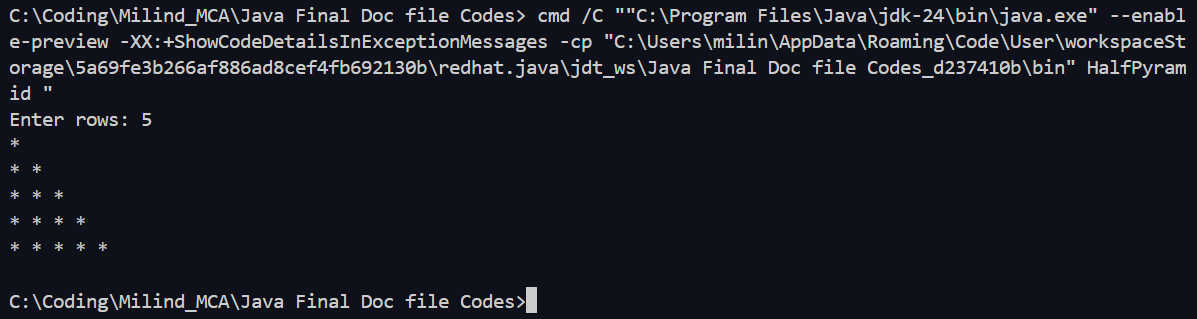
            System.out.println();

        }

    }

}

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No:2**

# ----------------------------------------------------------------

**AIM:- Defining a class. Creating an array of objects. Creating a package.**

**e.g create a class of complex numbers and perform addition on two numbers.**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

* **Complex.java**

package mymath;

public class Complex {

    double real, imaginary;

    public Complex(double r, double i) {

        real = r;

        imaginary = i;

    }

    public void add(Complex num) {

        real = real + num.real;

        imaginary = imaginary + num.imaginary;

    }

    public void subtraction(Complex num) {

        real = real - num.real;

        imaginary = imaginary - num.imaginary;

    }

    public void multiplication(Complex num) {

        double tempReal = (real \* num.real) - (imaginary \* num.imaginary);

        double tempImaginary = (real \* num.imaginary) + (imaginary \* num.real);

        real = tempReal;

        imaginary = tempImaginary;

    }

    public void display() {

        System.out.println(real + " + " + imaginary + "i");

    }

}

* **Main.java**

import mymath.Complex;

public class Main {

    public static void main(String[] args) {

        // Creating array of Complex objects

        Complex[] numbers = new Complex[2];

        numbers[0] = new Complex(3, 4);

        numbers[1] = new Complex(1, 2);

        // Perform addition

        numbers[0].add(numbers[1]);

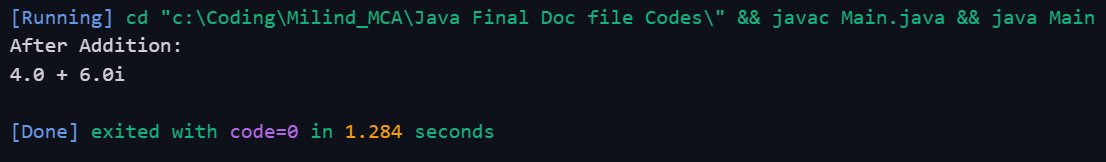
        System.out.println("After Addition:");

        numbers[0].display();

    }

}

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 3**

# ----------------------------------------------------------------

**AIM:- Write a Java program to create a superclass Vehicle having members Company and Price. Derive two different classes LightMotorVehicle (mileage) and HeavyMotorVehicle (capacity\_in\_tons). Accept the information for “n” vehicles and display the information in appropriate form. While taking data, ask the user about the type of vehicle first. (Inheritance & Interface).**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

import java.util.Scanner;

interface VehicleInfo {

    void displayInfo();

}

class Vehicle implements VehicleInfo {

    String company;

    double price;

    public Vehicle(String company, double price) {

        this.company = company;

        this.price = price;

    }

    @Override

    public void displayInfo() {

        System.out.println("Company: " + company);

        System.out.println("Price: " + price);

    }

}

class LightMotorVehicle extends Vehicle {

    double mileage;

    public LightMotorVehicle(String company, double price, double mileage) {

        super(company, price);

        this.mileage = mileage;

    }

    @Override

    public void displayInfo() {

        super.displayInfo();

        System.out.println("Mileage: " + mileage + " km/l");

    }

}

class HeavyMotorVehicle extends Vehicle {

    double capacityInTons;

    public HeavyMotorVehicle(String company, double price, double capacityInTons) {

        super(company, price);

        this.capacityInTons = capacityInTons;

    }

    @Override

    public void displayInfo() {

        super.displayInfo();

        System.out.println("Capacity: " + capacityInTons + " tons");

    }

}

public class VehicleDemo {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of vehicles: ");

        int n = scanner.nextInt();

        scanner.nextLine();

        Vehicle[] vehicles = new Vehicle[n];

        for (int i = 0; i < n; i++) {

            System.out.println("\nEnter information for Vehicle " + (i + 1));

            System.out.print("Enter vehicle type (1 for LightMotorVehicle, 2 for HeavyMotorVehicle): ");

            int vehicleType = scanner.nextInt();

            scanner.nextLine();

            System.out.print("Enter company name: ");

            String company = scanner.nextLine();

            System.out.print("Enter price of vehicle: ");

            double price = scanner.nextDouble();

            scanner.nextLine();

            if (vehicleType == 1) {

                System.out.print("Enter mileage (in km/l): ");

                double mileage = scanner.nextDouble();

                vehicles[i] = new LightMotorVehicle(company, price, mileage);

            } else if (vehicleType == 2) {

                System.out.print("Enter capacity in tons: ");

                double capacityInTons = scanner.nextDouble();

                vehicles[i] = new HeavyMotorVehicle(company, price, capacityInTons);

            } else {

                System.out.println("Invalid vehicle type! Please enter 1 or 2.");

                i--;

            }

            scanner.nextLine();

        }

        System.out.println("\nVehicle Information:");

        for (int i = 0; i < n; i++) {

            System.out.println("\nVehicle " + (i + 1) + " details:");

            vehicles[i].displayInfo();

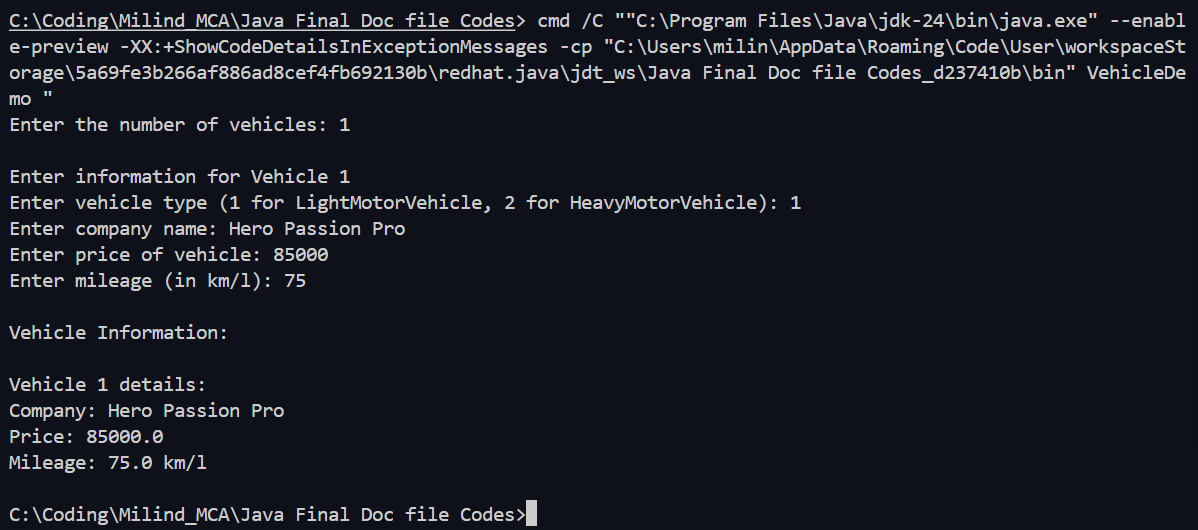
        }

        scanner.close();

    }

}

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 4**

# ----------------------------------------------------------------

**AIM:- Write a java program to create a User defined exception to check the following conditions and throw the exception if the criterion does not meet. a. User has age between 18 and 55 b. User stays has income between Rs. 50,000 – Rs. 1,00,000 per month c. User stays in Pune/ Mumbai/ Bangalore / Chennai d. User has 4-wheeler Accept age, Income, City, Vehicle from the user and check for the conditions mentioned above. If any of the conditions are not met then throw the exception.**

**(Inheritance & Interface)**

1. **useCard() method increases the creditAmount by a specific amount upto kreditLimit.**
2. **payCreadit() reduces the creditAmount by a specific amount.**
3. **increaseLimit() increases the credit Limit for GoldCardCustomers (only 3 times, not more than 5000 rupees each time.).**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

import java.util.Scanner;

interface CardOperations {

    void useCard(double amount);

    void payCredit(double amount);

    void increaseLimit(double amount);

}

class InvalidUserException extends Exception {

    public InvalidUserException(String message) {

        super(message);

    }

}

class User {

    int age;

    double income;

    String city;

    String vehicle;

    public void validateUser() throws InvalidUserException {

        if (age < 18 || age > 55)

            throw new InvalidUserException("Age must be between 18 and 55.");

        if (income < 50000 || income > 100000)

            throw new InvalidUserException("Income must be between Rs. 50,000 and Rs. 1,00,000.");

        if (!(city.equalsIgnoreCase("Pune") || city.equalsIgnoreCase("Mumbai") || city.equalsIgnoreCase("Nashik") ||

                city.equalsIgnoreCase("Bangalore") || city.equalsIgnoreCase("Chennai")))

            throw new InvalidUserException("City must be Nashik, Pune, Mumbai, Bangalore, or Chennai.");

        if (!(vehicle.equalsIgnoreCase("4-wheeler")))

            throw new InvalidUserException("User must have a 4-wheeler.");

    }

}

class GoldCardCustomer extends User implements CardOperations {

    double creditAmount = 0;

    double creditLimit = 20000;

    int limitIncreaseCount = 0;

    public void useCard(double amount) {

        if ((creditAmount + amount) <= creditLimit) {

            creditAmount += amount;

            System.out.println("Card used. Current credit amount: Rs. " + creditAmount);

        } else {

            System.out.println("Cannot use card. Limit exceeded.");

        }

    }

    public void payCredit(double amount) {

        creditAmount -= amount;

        if (creditAmount < 0)

            creditAmount = 0;

        System.out.println("Payment done. Remaining credit amount: Rs. " + creditAmount);

    }

    public void increaseLimit(double amount) {

        if (limitIncreaseCount < 3 && amount <= 5000) {

            creditLimit += amount;

            limitIncreaseCount++;

            System.out.println("Credit limit increased. New limit: Rs. " + creditLimit);

        } else {

            System.out.println("Limit cannot be increased further.");

        }

    }

}

public class Main2 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        GoldCardCustomer customer = new GoldCardCustomer();

        try {

            // Get user input

            System.out.print("Enter Age: ");

            customer.age = sc.nextInt();

            System.out.print("Enter Monthly Income: ");

            customer.income = sc.nextDouble();

            sc.nextLine(); // consume newline

            System.out.print("Enter City: ");

            customer.city = sc.nextLine();

            System.out.print("Enter Vehicle Type: ");

            customer.vehicle = sc.nextLine();

            customer.validateUser();

            System.out.println("User validated successfully!");

            customer.useCard(5000);

            customer.useCard(18000);

            customer.payCredit(2000);

            customer.increaseLimit(4000);

            customer.increaseLimit(3000);

            customer.increaseLimit(5000);

            customer.increaseLimit(1000);

        } catch (InvalidUserException e) {

            System.out.println("Validation Error: " + e.getMessage());

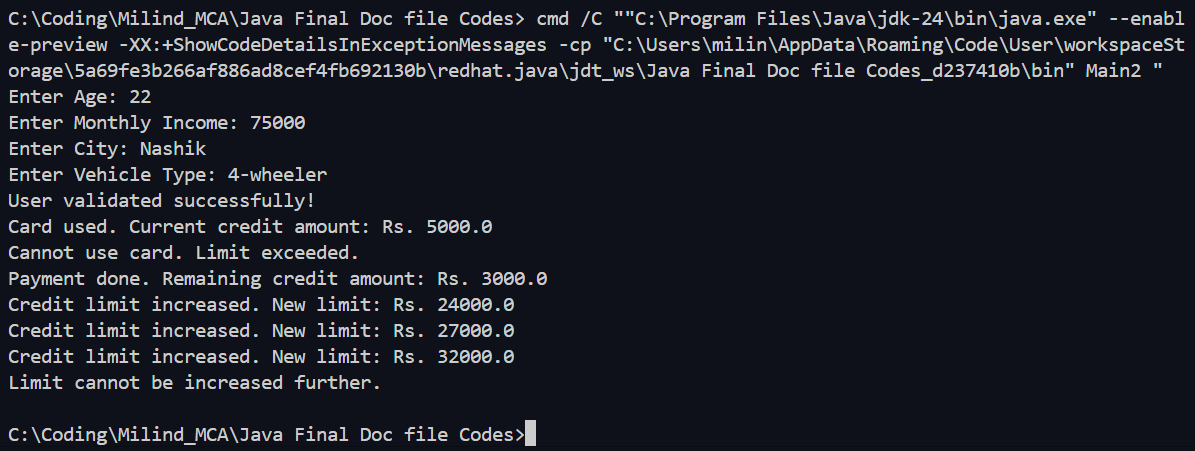
        }

        sc.close();

    }

}

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 5**

# ----------------------------------------------------------------

**AIM:- Accept N integers from the user and store them in a collection. Display them in the sorted order. The collection should not accept duplicate elements. (Use a suitable collection).**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

import java.util.Scanner;

import java.util.Set;

import java.util.TreeSet;

public class SortUniqueIntegers {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        // TreeSet: No duplicates + Sorted

        Set<Integer> numbers = new TreeSet<>();

        System.out.print("Enter how many numbers you want to enter (N): ");

        int n = sc.nextInt();

        System.out.println("Enter " + n + " integers:");

        for (int i = 0; i < n; i++) {

            int num = sc.nextInt();

            numbers.add(num); // Duplicate values will be ignored automatically

        }

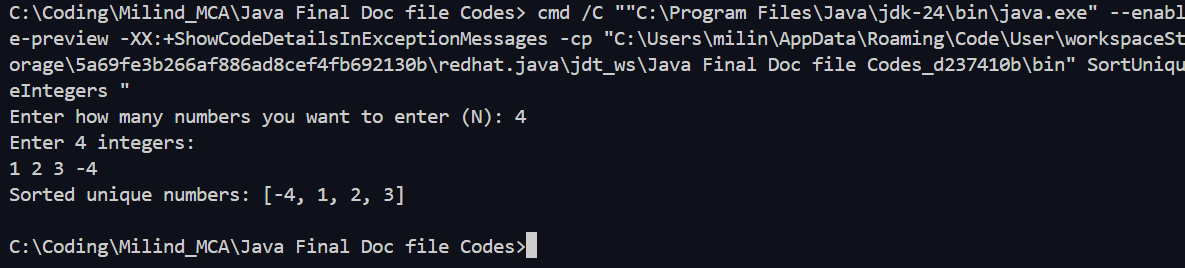
        System.out.println("Sorted unique numbers: " + numbers);

        sc.close();

    }

}

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 6**

# ----------------------------------------------------------------

**AIM:- Create a Hash table containing Student Name and Percentage. Display the contents of the hash table. Also search for a specific Student and display his percentage.**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

import java.util.Hashtable;

import java.util.Scanner;

public class StudentHashTable {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        Hashtable<String, Double> studentTable = new Hashtable<>();

        System.out.print("Enter number of students: ");

        int n = sc.nextInt();

        sc.nextLine(); // consume newline

        // Input: Student Name and Percentage

        for (int i = 0; i < n; i++) {

            System.out.print("Enter student name: ");

            String name = sc.nextLine();

            System.out.print("Enter percentage: ");

            double percentage = sc.nextDouble();

            sc.nextLine(); // consume newline

            studentTable.put(name, percentage);

        }

        // Display all entries

        System.out.println("\n--- Student Records ---");

        for (String key : studentTable.keySet()) {

            System.out.println("Name: " + key + ", Percentage: " + studentTable.get(key));

        }

        // Search for a student

        System.out.print("\nEnter name of student to search: ");

        String searchName = sc.nextLine();

        if (studentTable.containsKey(searchName)) {

            System.out.println("Percentage of " + searchName + ": " + studentTable.get(searchName));

        } else {

            System.out.println("Student " + searchName + " not found.");

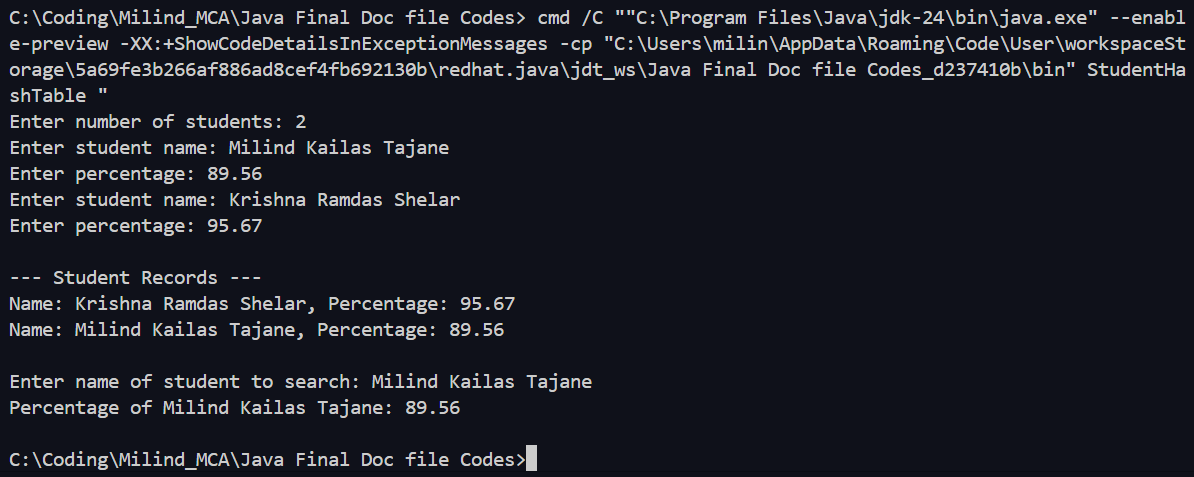
        }

        sc.close();

    }

}

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 7**

# ----------------------------------------------------------------

**AIM:- Write a Java program that reads a list of integers from the user and throws an exception if any numbers are duplicates.**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

import java.util.\*;

class DuplicateNumberException extends Exception {

    public DuplicateNumberException(String message) {

        super(message);

    }

}

public class DuplicateCheck {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        Set<Integer> numberSet = new HashSet<>();

        System.out.print("Enter how many numbers: ");

        int n = sc.nextInt();

        System.out.println("Enter " + n + " integers:");

        try {

            for (int i = 0; i < n; i++) {

                int num = sc.nextInt();

                if (!numberSet.add(num)) {

                    // If add() returns false, it means the number already exists

                    throw new DuplicateNumberException("Duplicate number found: " + num);

                }

            }

            System.out.println("All numbers are unique: " + numberSet);

        } catch (DuplicateNumberException e) {

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

        }

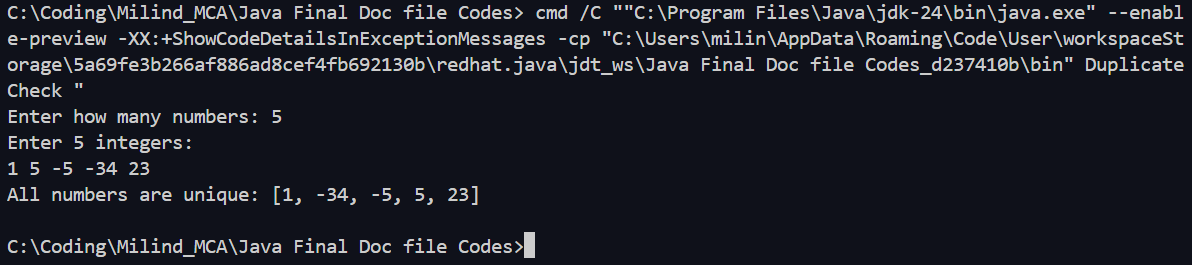
        sc.close();

    }

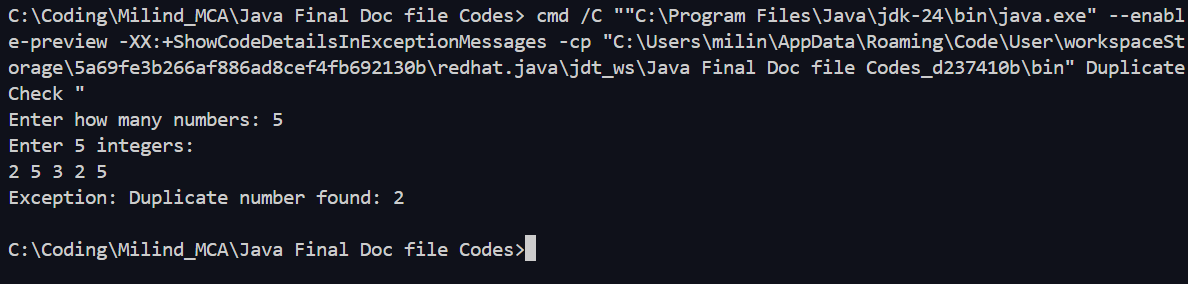
}

**Output:-**

**Case with no duplicates:**



**Case with duplicates:**

****

**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 8**

# ----------------------------------------------------------------

**AIM:- Write a Java program to replace the second element of an ArrayList with the specified element.**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

import java.util.ArrayList;

import java.util.Scanner;

public class ReplaceSecondElement {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        ArrayList<String> list = new ArrayList<>();

        System.out.print("Enter number of elements in ArrayList: ");

        int n = sc.nextInt();

        sc.nextLine();

        System.out.println("Enter " + n + " elements:");

        for (int i = 0; i < n; i++) {

            list.add(sc.nextLine());

        }

        if (list.size() >= 2) {

            System.out.print("Enter element to replace second element: ");

            String newElement = sc.nextLine();

            list.set(1, newElement);

            System.out.println("Updated ArrayList: " + list);

        } else {

            System.out.println("ArrayList must have at least 2 elements.");

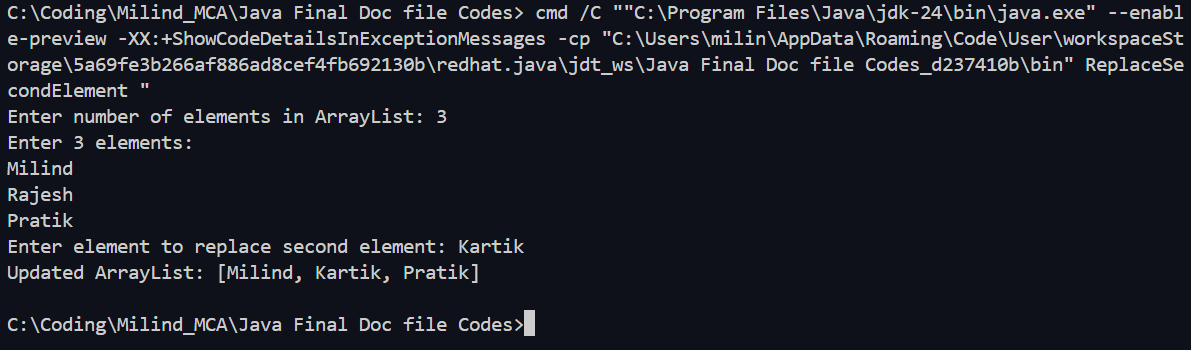
        }

        sc.close();

    }

}

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 9**

# ----------------------------------------------------------------

**AIM:- Write a Java program that demonstrates the use of the synchronized keyword in a multithreading scenario. For example, implement a file copying program where one thread reads data from a file while another thread writes the data to a new file, ensuring proper synchronization between the threads.**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

import java.io.\*;

class SharedData {

    String line;

    boolean hasData = false;

    // Writer waits until reader puts data

    public synchronized void write(String line) {

        while (hasData) {

            try {

                wait();

            } catch (InterruptedException e) {

            }

        }

        this.line = line;

        hasData = true;

        notify();

    }

    // Reader waits until writer reads data

    public synchronized String read() {

        while (!hasData) {

            try {

                wait();

            } catch (InterruptedException e) {

            }

        }

        hasData = false;

        notify();

        return line;

    }

}

class ReaderThread extends Thread {

    SharedData data;

    BufferedReader br;

    ReaderThread(SharedData data, String inputFile) throws IOException {

        this.data = data;

        br = new BufferedReader(new FileReader(inputFile));

    }

    public void run() {

        try {

            String line;

            while ((line = br.readLine()) != null) {

                data.write(line); // Send to writer

            }

            data.write("EOF"); // End marker

            br.close();

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

}

class WriterThread extends Thread {

    SharedData data;

    BufferedWriter bw;

    WriterThread(SharedData data, String outputFile) throws IOException {

        this.data = data;

        bw = new BufferedWriter(new FileWriter(outputFile));

    }

    public void run() {

        try {

            String line;

            while (!(line = data.read()).equals("EOF")) {

                bw.write(line);

                bw.newLine();

            }

            bw.close();

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

}

public class SimpleFileCopySync {

    public static void main(String[] args) {

        try {

            SharedData shared = new SharedData();

            // Change these to your actual file paths

            ReaderThread reader = new ReaderThread(shared, "input.txt");

            WriterThread writer = new WriterThread(shared, "output.txt");

            reader.start();

            writer.start();

            reader.join();

            writer.join();

            System.out.println("File copied successfully using synchronized threads!");

        } catch (Exception e) {

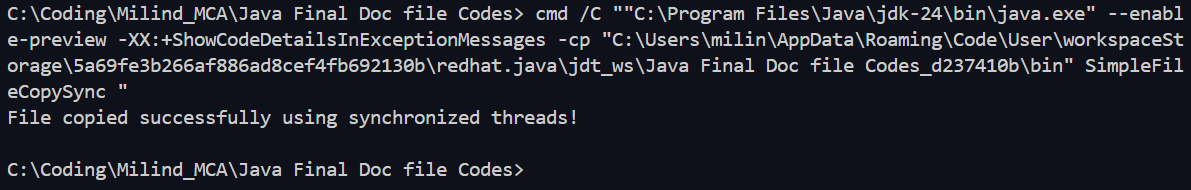
            e.printStackTrace();

        }

    }

}

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 10**

# ----------------------------------------------------------------

**AIM:- .Write a java program to Convert decimal to binary, octal, hex.**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

import java.util.Scanner;

class NumberConverter {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a decimal number: ");

        int decimal = scanner.nextInt();

        System.out.println("Binary: " + Integer.toBinaryString(decimal));

        System.out.println("Octal: " + Integer.toOctalString(decimal));

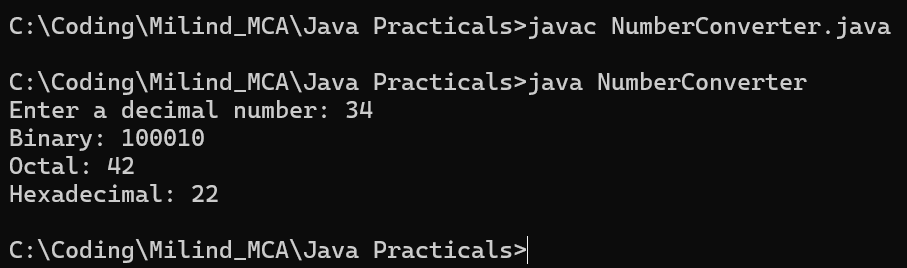
        System.out.println("Hexadecimal: " + Integer.toHexString(decimal).toUpperCase());

        scanner.close();

    }

}

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 11**

# ----------------------------------------------------------------

**AIM:- Write a java program to Print prime numbers 1 to 100.**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

class PrimeNumbers {

    public static void main(String[] args) {

        System.out.println("Prime numbers from 1 to 100:");

        for (int num = 2; num <= 100; num++) {

            if (isPrime(num)) {

                System.out.print(num + " ");

            }

        }

    }

    public static boolean isPrime(int n) {

        if (n < 2) return false;

        for (int i = 2; i <= Math.sqrt(n); i++) {

            if (n % i == 0) return false;

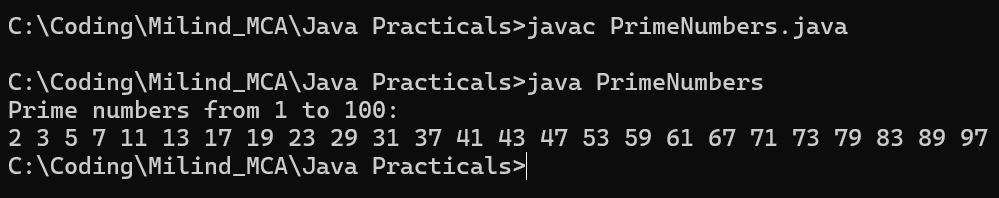
        }

        return true;

    }

}

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 12**

# ----------------------------------------------------------------

**AIM:- Write a java program to print Factorial of a number.**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

import java.util.Scanner;

class Factorial {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");

        int num = scanner.nextInt();

        System.out.println("Factorial of " + num + " is: " + factorial(num));

        scanner.close();

    }

    public static long factorial(int n) {

        long fact = 1;

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

            fact \*= i;

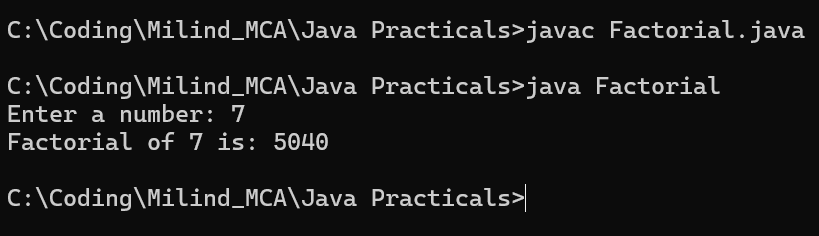
        }

        return fact;

    }

}

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 13**

# ----------------------------------------------------------------

**AIM:- Write a java program to print the pattern.**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

import java.util.Scanner;

class PatternPrinter {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of rows: ");

        int rows = scanner.nextInt();

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

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

                System.out.print("\* ");

            }

            System.out.println();

        }

        scanner.close();

    }

}

**Output:-**

