Module 2) Core Java

- Write a Java program to Take three numbers from the user and print the greatest number.

Main.java

```java
import java.util.Scanner;

class HelloWorld {
    public static void main(String[] args) {
        // Declare three variables
        int a,b,c;

        // scanner class
        Scanner scan = new Scanner(System.in);

        // Take 3 numbers from stdin
        System.out.println("Enter number 1: ");
        a = scan.nextInt();
        System.out.println("Enter number 2: ");
        b = scan.nextInt();
        System.out.println("Enter number 3: ");
        c = scan.nextInt();


        // Do logical comparison
        if(a > b && a > c){
            System.out.println("A is greater ");
        }
        else if(b > a && b > c){
            System.out.println("B is greater ");
        }
        else if(c > a && c > b){
            System.out.println("C is greater ");
        }
    }
}
```

Output

```
java -cp /tmp/e7OAIIzPfH/HelloWorld
Enter number 1:
1
Enter number 2:
2
Enter number 3:
3
C is greater

=== Code Execution Successful ===
```

- Write a Java program that takes the user to provide a single character from the alphabet. Print Vowel or Consonant, depending on the user input. If the user input is not a letter (between a and z or A and Z), or is a string of length > 1, print an error message.

Main.java

```java
import java.util.Scanner;

public class VowelOrConsonant {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Prompt the user to enter a single charachter
        System.out.print("Enter a single character from the alphabet: ");
        String input = scanner.next();

        // Check if the input is not a single letter
        if (input.length() != 1 || !Character.isLetter(input.charAt(0))) {
            System.out.println("Error: Please enter a single alphabet character.");
        } else {
            // Convert the character to lower case
            char ch = Character.toLowerCase(input.charAt(0));
            // Check if the character is a vowel
            switch (ch) {
                case 'a':
                case 'e':
                case 'i':
                case 'o':
                case 'u':
                    System.out.println("Vowel");
                    break;
                default:
                    System.out.println("Consonant");
            }
        }

        // Close the scanner
        scanner.close();
    }
}
```

Output

```
java -cp /tmp/iEzcurknfs/VowelOrConsonant
Enter a single character from the alphabet: a
Vowel

=== Code Execution Successful ===
```

- Write a Java program that takes a year from user and print whether that year is a leap year or not. B19. Write a program in Java to display the first 10 natural numbers using while loop.

Main.java

```java
import java.util.Scanner;

public class LeapYear {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Prompt the user to enter a year
        System.out.print("Enter a year: ");
        int year = scanner.nextInt();

        // Check if the year is a leap year
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
            System.out.println(year + " is a leap year.");
        } else {
            System.out.println(year + " is not a leap year.");
        }

        // Close the scanner
        scanner.close();
    }
}
```

Output

```
java -cp /tmp/nXiBwlOqbf/LeapYear
Enter a year: 2000
2000 is a leap year.

=== Code Execution Successful ===
```

-
- Write a program in Java to input 5 numbers from keyboard and find their sum and average using for loop.

Main.java

```java
public class FirstTenNaturalNumbers {
    public static void main(String[] args) {
        int i = 1;

        // Use a while loop to print the first 10 natural numbers
        while (i <= 10) {
            System.out.println(i);
            i++;
        }
    }
}
```

Output

```
java -cp /tmp/rmCtQz8jiG/FirstTenNaturalNumbers
1
2
3
4
5
6
7
8
9
10

=== Code Execution Successful ===
```

- Write a programin Java to display the pattern like right angle triangle with a number.

Main.java

```java
public class NumberTriangle {
    public static void main(String[] args) {
        int rows = 5;

        // Use nested loops to print the number triangle
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(j);
            }
            // Move to the next line after printing each row
            System.out.println();
        }
    }
}
```

Output

```
java -cp /tmp/iByt3YQ9TD/NumberTriangle
1
12
123
1234
12345

=== Code Execution Successful ===
```

• Write a program in Java to make such a pattern like right angle triangle with number increased by 1 The pattern like:

Main.java

```java
public class NumberTriangleIncrement {
    public static void main(String[] args) {
        int rows = 5;
        int num = 1;

        // Use nested loops to print the number triangle
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                // Print the current number
                System.out.print(num + " ");
                num++;
            }
            // Move to the next line after printing each row
            System.out.println();
        }
    }
}
```

Output

```
java -cp /tmp/irXcy9Dg9y/NumberTriangleIncrement
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

=== Code Execution Successful ===
```

• Write a Java program that reads a positive integer and count the number of digits of the number. Input an integer number less than ten billion: 125463 Number of digits in the number: 6

Main.java

```java
import java.util.Scanner;

public class DigitCounter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Prompt the user to enter a positive integer
        System.out.print("Input an integer number less than ten billion: ");
        long number = scanner.nextLong();

        // Count the number of digits in the number
        int count = 0;
        long temp = number;
        while (temp != 0) {
            temp /= 10;
            count++;
        }

        System.out.println("Number of digits in the number: " + count);

        scanner.close();
    }
}
```

Output

```
java -cp /tmp/LhyKXyimMz/DigitCounter
Input an integer number less than ten billion: 1293012
Number of digits in the number: 7

=== Code Execution Successful ===
```

• Write a Java program to count the letters, spaces, numbers and other characters of an input string.

Main.java

```java
import java.util.Scanner;

public class CharacterCounter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Prompt the user to enter a string
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        int letters = 0, spaces = 0, numbers = 0, others = 0;

        // Count the characters
        for (char ch : input.toCharArray()) {
            if (Character.isLetter(ch)) {
                letters++;
            } else if (Character.isDigit(ch)) {
                numbers++;
            } else if (Character.isSpaceChar(ch)) {
                spaces++;
            } else {
                others++;
            }
        }

        System.out.println("Letters: " + letters);
        System.out.println("Spaces: " + spaces);
        System.out.println("Numbers: " + numbers);
        System.out.println("Other characters: " + others);

        scanner.close();
    }
}
```

Output

```
java -cp /tmp/qGR7tbEEaW/CharacterCounter
Enter a string: hello java world
Letters: 14
Spaces: 2
Numbers: 0
Other characters: 0

=== Code Execution Successful ===
```

• Write a Java program to print the ASCII value of a given character.

Main.java

```java
import java.util.Scanner;

public class ASCIIValue {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Prompt the user to enter a character
        System.out.print("Enter a character: ");
        char ch = scanner.next().charAt(0);

        // Print the ASCII value
        int ascii = (int) ch;
        System.out.println("The ASCII value of '" + ch + "' is: " +
            ascii);

        scanner.close();
    }
}
```

Output

```
java -cp /tmp/P335oRHUkN/ASCIIValue
Enter a character: hello java
The ASCII value of 'h' is: 104

=== Code Execution Successful ===
```

• Write a Java program that accepts an integer (n) and computes the value of n+nn+nnn.
Input number: 5 5 + 55 + 555

Main.java

```java
import java.util.Scanner;

public class ComputeValue {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Input number: ");
        int n = scanner.nextInt();

        int nn = Integer.parseInt("" + n + n);
        int nnn = Integer.parseInt("" + n + n + n);

        int result = n + nn + nnn;
        System.out.println(n + " + " + nn + " + " + nnn + " = " +
            result);

        scanner.close();
    }
}
```

Output

```
java -cp /tmp/Zx65BtYkti/ComputeValue
Input number: 10
10 + 1010 + 101010 = 102030

=== Code Execution Successful ===
```

• Write a Java program to display the system time.

Main.java

```java
import java.util.Date;

public class SystemTime {
    public static void main(String[] args) {
        Date date = new Date();
        System.out.println("Current system time: " + date.toString());
    }
}
```

Output

```
java -cp /tmp/oZOPhXX6Au/SystemTime
Current system time: Mon Aug 05 11:36:15 GMT 2024

=== Code Execution Successful ===
```

• Write a Java program to print numbers between 1 to 100 which are divisible by 3, 5 and by both.

Main.java

```java
public class DivisibleNumbers {
    public static void main(String[] args) {
        System.out.println("Numbers divisible by 3:");
        for (int i = 1; i <= 100; i++) {
            if (i % 3 == 0) {
                System.out.print(i + " ");
            }
        }

        System.out.println("\nNumbers divisible by 5:");
        for (int i = 1; i <= 100; i++) {
            if (i % 5 == 0) {
                System.out.print(i + " ");
            }
        }

        System.out.println("\nNumbers divisible by both 3 and 5:");
        for (int i = 1; i <= 100; i++) {
            if (i % 3 == 0 && i % 5 == 0) {
                System.out.print(i + " ");
            }
        }
    }
}
```

Output

```
java -cp /tmp/xoBPm3NWVL/DivisibleNumbers
Numbers divisible by 3:
3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60 63 66 69 72 75 78 81 84 87
    90 93 96 99
Numbers divisible by 5:
5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
Numbers divisible by both 3 and 5:
15 30 45 60 75 90
=== Code Execution Successful ===
```

• W.A.J.P to get the character at the given index within the String. Original String = Tops Technologies! The character at position 0 is T, The character at position 10 is o

Main.java

```java
import java.util.Scanner;

public class CharacterAtIndex {
    public static void main(String[] args) {
        String originalString = "Tops Technologies!";
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the index: ");
        int index = scanner.nextInt();

        if (index < 0 || index >= originalString.length()) {
            System.out.println("Invalid index.");
        } else {
            char charAtPos = originalString.charAt(index);
            System.out.println("The character at position " + index +
                " is " + charAtPos);
        }

        scanner.close();
    }
}
```

Output

```
java -cp /tmp/kRsYThMhj6/CharacterAtIndex
Enter the index: 10
The character at position 10 is o

=== Code Execution Successful ===
```

• W.A.J.P to concatenate a given string to the end of another string.

Main.java

```java
import java.util.Scanner;

public class ConcatenateStrings {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter first string: ");
        String str1 = scanner.nextLine();
        System.out.print("Enter second string: ");
        String str2 = scanner.nextLine();

        String concatenatedString = str1 + str2;
        System.out.println("Concatenated string: " +
            concatenatedString);

        scanner.close();
    }
}
```

Output

```
java -cp /tmp/rbRRUxcezo/ConcatenateStrings
Enter first string: Tops
Enter second string: Technologies
Concatenated string: TopsTechnologies

=== Code Execution Successful ===
```

• W.A.J.P to compare a given string to the specified character sequence. Comparing topsint.com and topsint.com: true Comparing Topsint.com and topsint.com: false

Main.java

```java
import java.util.Scanner;

public class CompareStrings {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter first string: ");
        String str1 = scanner.nextLine();
        System.out.print("Enter second string: ");
        String str2 = scanner.nextLine();

        boolean areEqual = str1.equals(str2);
        System.out.println("Comparing " + str1 + " and " + str2 + ": "
            + areEqual);

        scanner.close();
    }
}
```

Output

```
java -cp /tmp/dGSvBD6RUy/CompareStrings
Enter first string: tops
Enter second string: Tops
Comparing tops and Tops: false

=== Code Execution Successful ===
```

• W.A.J.P to check whether a given string ends with the contents of another string. "Java Exercises" ends with "se"? False "Java Exercise" ends with "se"? True

Main.java

```java
import java.util.Scanner;

public class StringEndsWith {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the main string: ");
        String mainString = scanner.nextLine();
        System.out.print("Enter the substring to check: ");
        String subString = scanner.nextLine();

        boolean endsWith = mainString.endsWith(subString);
        System.out.println("\"" + mainString + "\" ends with \"" +
            subString + "\"? " + endsWith);

        scanner.close();
    }
}
```

Output

```
java -cp /tmp/VVo43Xym85/StringEndsWith
Enter the main string: test
Enter the substring to check: st
"test" ends with "st"? true

=== Code Execution Successful ===
```

• W.A.J.P to check whether a given string starts with the contents of another string. Red is favorite color. Starts with Red? True Orange is also my favorite color. Starts with Red? False I3.

Main.java

```java
import java.util.Scanner;

public class StringStartsWith {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the main string: ");
        String mainString = scanner.nextLine();
        System.out.print("Enter the substring to check: ");
        String subString = scanner.nextLine();

        boolean startsWith = mainString.startsWith(subString);
        System.out.println("\"" + mainString + "\" starts with \"" +
            subString + "\"? " + startsWith);

        scanner.close();
    }
}
```

Output

```
java -cp /tmp/a7jCPdDXX6/StringStartsWith
Enter the main string: hello
Enter the substring to check: hell
"hello" starts with "hell"? true

=== Code Execution Successful ===
```

• W.A.J.P to find all interleaving of given strings. The given strings are: WX YZ The interleaving strings are: YWZX WYZX YWXZ WXYZ YZWX WYXZ

Main.java

```java
import java.util.HashSet;
import java.util.Set;

public class StringInterleavings {
    public static void main(String[] args) {
        String str1 = "WX";
        String str2 = "YZ";

        Set<String> result = new HashSet<>();
        interleave(result, str1, str2, "");
        System.out.println("The interleaving strings are: " + result);
    }

    private static void interleave(Set<String> result, String str1, String str2, String
        interleaved) {
        if (str1.isEmpty() && str2.isEmpty()) {
            result.add(interleaved);
            return;
        }
        if (!str1.isEmpty()) {
            interleave(result, str1.substring(1), str2, interleaved + str1.charAt(0));
        }
        if (!str2.isEmpty()) {
            interleave(result, str1, str2.substring(1), interleaved + str2.charAt(0));
        }
    }
}
```

Output

```
java -cp /tmp/68pfPL6hnK/StringInterleavings
The interleaving strings are: [YWZX, WYZX, YWXZ, WXYZ, YZWX, WYXZ]

=== Code Execution Successful ===
```

• W.A.J.P to find the second most frequent character in a given string. The given string is: successes The second most frequent char in the string is: c

Main.java

```java
import java.util.HashMap;
import java.util.Map;

public class SecondMostFrequent {
    public static void main(String[] args) {
        String str = "successes";
        char secondMostFrequent = findSecondMostFrequentChar(str);
        System.out.println("The second most frequent char in the string is: " + secondMostFrequent);
    }

    private static char findSecondMostFrequentChar(String str) {
        Map<Character, Integer> charCountMap = new HashMap<>();
        for (char c : str.toCharArray()) {
            charCountMap.put(c, charCountMap.getOrDefault(c, 0) + 1);
        }

        char mostFrequent = ' ';
        char secondMostFrequent = ' ';
        int maxCount = 0, secondMaxCount = 0;

        for (Map.Entry<Character, Integer> entry : charCountMap.entrySet()) {
            int count = entry.getValue();
            if (count > maxCount) {
                secondMaxCount = maxCount;
                secondMostFrequent = mostFrequent;
                maxCount = count;
                mostFrequent = entry.getKey();
            } else if (count > secondMaxCount) {
                secondMaxCount = count;
                secondMostFrequent = entry.getKey();
            }
        }

        return secondMostFrequent;
    }
}
```

Output

```
java -cp /tmp/TphxIXFmDz/SecondMostFrequent
The second most frequent char in the string is: c

=== Code Execution Successful ===
```

• Create a class named 'Print Number' to print various numbers of different data types by creating different methods with the same name 'printn' having a parameter for each data type.

Main.java

```java
class PrintNumber {
    // Method to print an integer
    void printn(int number) {
        System.out.println("Integer: " + number);
    }

    // Method to print a double
    void printn(double number) {
        System.out.println("Double: " + number);
    }

    // Method to print a float
    void printn(float number) {
        System.out.println("Float: " + number);
    }

    // Method to print a long
    void printn(long number) {
        System.out.println("Long: " + number);
    }

    // Method to print a short
    void printn(short number) {
        System.out.println("Short: " + number);
    }

    // Method to print a byte
    void printn(byte number) {
        System.out.println("Byte: " + number);
    }

    // Method to print a character
    void printn(char character) {
        System.out.println("Character: " + character);
    }
}

public class TestPrintNumber {
    public static void main(String[] args) {
        PrintNumber printer = new PrintNumber();

        printer.printn(5);
        printer.printn(5.5);
        printer.printn(5.5f);
        printer.printn(500000L);
        printer.printn((short) 5);
        printer.printn((byte) 5);
        printer.printn('A');
    }
}
```

Output

```
java -cp /tmp/vf5IXqYKni/TestPrintNumber
Integer: 5
Double: 5.5
Float: 5.5
Long: 500000
Short: 5
Byte: 5
Character: A

=== Code Execution Successful ===
```

• Create a class to print an integer and a character with two methods having the same name but different sequence of the integer and the character parameters. For example, if the parameters of the first method are of the form (int n, char c), then that of the second method will be of the form (char c, int n).

Main.java

```java
class Example {
    // Methd with paramters (int n, char c)
    void method(int n, char c) {
        System.out.println("Method with int and char: " + n + ", " + c);
    }

    // Methd with paramters (char c, int n)
    void method(char c, int n) {
        System.out.println("Method with char and int: " + c + ", " + n);
    }
}

public class TestExample {
    public static void main(String[] args) {
        Example example = new Example();
        example.method(5, 'A');
        example.method('B', 10);
    }
}
```

Output

```
java -cp /tmp/Vibe4XUcSP/TestExample
Method with int and char: 5, A
Method with char and int: B, 10

=== Code Execution Successful ===
```

• Create a class to print the area of a square and a rectangle. The class has two methods with the same name but different number of parameters. The method for printing area of a rectangle has two parameters which are length and breadth respectively while the other method for printing area of square has one parameter which is side of square.

Main.java

```java
class Area {
    // Methd to print area of square
    void printArea(int side) {
        int area = side * side;
        System.out.println("Area of square: " + area);
    }

    // Methd to print area of rectangle
    void printArea(int length, int breadth) {
        int area = length * breadth;
        System.out.println("Area of rectangle: " + area);
    }
}

public class TestArea {
    public static void main(String[] args) {
        Area area = new Area();
        area.printArea(5); // Print area of square
        area.printArea(5, 10); // Print area of rectangle
    }
}
```

Output

```
java -cp /tmp/g2U23WysPE/TestArea
Area of square: 25
Area of rectangle: 50

=== Code Execution Successful ===
```

• Create a class with a method that prints "This is a parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call 1 - method of parent class by object of parent class 2 - method of child class by object of child class 3 - method of parent class by object of child class

Main.java

```java
class Parent {
    // Methd to print message from parent class
    void message() {
        System.out.println("This is a parent class");
    }
}

class Child extends Parent {
    // Methd to print message from child class
    void message() {
        System.out.println("This is a child class");
    }
}

public class TestParentChild {
    public static void main(String[] args) {
        Parent parent = new Parent();
        Child child = new Child();

        parent.message(); // Method of parent class by object of parent class
        child.message(); // Method of child class by object of child class
        ((Parent) child).message(); // Method of parent class by object of
            child class
    }
}
```

Output

```
java -cp /tmp/nmMLLrZ7FU/TestParentChild
This is a parent class
This is a child class
This is a child class

=== Code Execution Successful ===
```

• Create a class named 'Member' having the following members: 1. Data members 2. Name 3. Age 4. Phone number 5. Address 6. Salary It also has a method named 'printSalary' which prints the salary of the members.

Main.java

```java
class Member {
    String name;
    int age;
    String phoneNumber;
    String address;
    double salary;

    // Methd to print salary
    void printSalary() {
        System.out.println("Salary: " + salary);
    }
}

public class TestMember {
    public static void main(String[] args) {
        Member member = new Member();
        member.name = "John Doe";
        member.age = 30;
        member.phoneNumber = "123-456-7890";
        member.address = "123 Main St";
        member.salary = 50000.0;

        member.printSalary(); // Print the salary
    }
}
```

Output

```
java -cp /tmp/GrwRYRtRnv/TestMember
Salary: 50000.0

=== Code Execution Successful ===
```

• Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize the length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super (s, s)'. Print the area and perimeter of a rectangle and a square.

Main.java

```java
class Rectangle {
    int length;
    int breadth;

    // Constructor to initialize length and breadth
    Rectangle(int length, int breadth) {
        this.length = length;
        this.breadth = breadth;
    }

    // Methd to print area
    void printArea() {
        int area = length * breadth;
        System.out.println("Area of rectangle: " + area);
    }

    // Methd to print perimeter
    void printPerimeter() {
        int perimeter = 2 * (length + breadth);
        System.out.println("Perimeter of rectangle: " + perimeter);
    }
}

class Square extends Rectangle {
    // Constructor for square
    Square(int side) {
        super(side, side);
    }
}

public class TestRectangleSquare {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(5, 10);
        Square square = new Square(5);

        rectangle.printArea(); // Print area of rectangle
        rectangle.printPerimeter(); // Print perimeter of rectangle

        square.printArea(); // Print area of square
        square.printPerimeter(); // Print perimeter of square
    }
}
```

Output

```
java -cp /tmp/Z7PXJd8kDZ6/TestRectangleSquare
Area of rectangle: 50
Perimeter of rectangle: 30
Area of rectangle: 25
Perimeter of rectangle: 20

=== Code Execution Successful ===
```

• Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' without any parameter in its constructor.

Main.java

```java
class Triangle {
    int a = 3;
    int b = 4;
    int c = 5;

    // Methd to print area
    void printArea() {
        double s = (a + b + c) / 2.0;
        double area = Math.sqrt(s * (s - a) * (s - b) * (s - c));
        System.out.println("Area of triangle: " + area);
    }

    // Methd to print perimeter
    void printPerimeter() {
        int perimeter = a + b + c;
        System.out.println("Perimeter of triangle: " + perimeter);
    }
}

public class TestTriangle {
    public static void main(String[] args) {
        Triangle triangle = new Triangle();
        triangle.printArea(); // Print area of triangle
        triangle.printPerimeter(); // Print perimeter of triangle
    }
}
```

Output

```
java -cp /tmp/5UZZPh1gLJ/TestTriangle
Area of triangle: 6.0
Perimeter of triangle: 12

=== Code Execution Successful ===
```

• Print the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate methods for each operation whose real and imaginary parts are entered by user.

Main.java

```java
import java.util.Scanner;

class Complex {
    double real;
    double imaginary;

    // Constructor to initialize complex number
    Complex(double real, double imaginary) {
        this.real = real;
        this.imaginary = imaginary;
    }

    // Methd to add two complex numbers
    Complex add(Complex other) {
        return new Complex(this.real + other.real, this.imaginary +
            other.imaginary);
    }

    // Methd to subtract two complex numbers
    Complex subtract(Complex other) {
        return new Complex(this.real - other.real, this.imaginary -
```

Output

```
java -cp /tmp/d2iEMaHHb9/TestComplex
Enter real part of first complex number: 12
Enter imaginary part of first complex number: 12
Enter real part of second complex number: 123
Enter imaginary part of second complex number: 123
Sum: 135.0 + 135.0i
Difference: -111.0 + -111.0i
Product: 0.0 + 2952.0i

=== Code Execution Successful ===1
```

• Create an abstract class 'Parent' with a method 'message'. It has two subclasses each having a method with the same name 'message' that prints "This is first subclass" and "This is second subclass" respectively. Call the methods 'message' by creating an object for each subclass.

Main.java

```java
abstract class Parent {
    // Methd to print message
    abstract void message();
}

class FirstSubclass extends Parent {
    // Methd to print message from first subclass
    void message() {
        System.out.println("This is first subclass");
    }
}

class SecondSubclass extends Parent {
    // Methd to print message from second subclass
    void message() {
        System.out.println("This is second subclass");
    }
}

public class TestAbstractClass {
    public static void main(String[] args) {
        Parent obj1 = new FirstSubclass();
```

Output

```
java -cp /tmp/pP6X1U0v0s/TestAbstractClass
This is first subclass
This is second subclass

=== Code Execution Successful ===
```

• Create an abstract class 'Bank' with an abstract method 'getBalance'. $100, $150 and $200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.

Main.java

```java
abstract class Bank {
    // Abstact method to get balance
    abstract int getBalance();
}

class BankA extends Bank {
    // Return balance for Bank A
    int getBalance() {
        return 100;
    }
}

class BankB extends Bank {
    // Return balance for Bank B
    int getBalance() {
        return 150;
    }
}

class BankC extends Bank {
    // Return balance for Bank C
    int getBalance() {
        return 200;
    }
}

public class TestBank {
    public static void main(String[] args) {
        Bank bankA = new BankA();
        Bank bankB = new BankB();
        Bank bankC = new BankC();
```

Output

```
java -cp /tmp/S2XH3pR2cz/TestBank
Balance in Bank A: $100
Balance in Bank B: $150
Balance in Bank C: $200

=== Code Execution Successful ===
```

• We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects asits parameters and the marks in foursubjects as its parameters forstudent B. Create an object for each of the two classes and print the percentage of marks for both the students.

Main.java

```java
abstract class Marks {
    // Abstact method to get percentage
    abstract double getPercentage();
}

class A extends Marks {
    private int marks1, marks2, marks3;

    // Constructor for student A
    A(int marks1, int marks2, int marks3) {
        this.marks1 = marks1;
        this.marks2 = marks2;
        this.marks3 = marks3;
    }

    // Calculate and return percentage for student A
    double getPercentage() {
        return (marks1 + marks2 + marks3) / 3.0;
    }
}

class B extends Marks {
    private int marks1, marks2, marks3, marks4;

    // Constructor for student B
    B(int marks1, int marks2, int marks3, int marks4) {
        this.marks1 = marks1;
        this.marks2 = marks2;
        this.marks3 = marks3;
        this.marks4 = marks4;
    }
```

Output

```
java -cp /tmp/HGcDR9cfId/TestMarks
Percentage of student A: 90.0%
Percentage of student B: 87.5%

=== Code Execution Successful ===
```

• Write a program to print the factorial of a number by defining a method named 'Factorial'. Factorial of any number n is represented by n! And is equal to 1*2*3*. *(n-1) *n. E.g.- 4! = 1*2*3*4 = 24 3! = 3*2*1 = 6 2! = 2*1 = 2 Also, 1! = 1 0! = 0

Main.java

```java
import java.util.Scanner;

public class Factorial {
    // Method to calculate factorial
    static int factorial(int n) {
        if (n == 0) {
            return 1;
        }
        return n * factorial(n - 1);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        // Calculate and print factorial
        System.out.println("Factorial of " + number + " is " +
            factorial(number));

        scanner.close();
    }
```

Output

```
java -cp /tmp/4h4aZvuVhu/Factorial
Enter a number: 10
Factorial of 10 is 3628800

=== Code Execution Successful ===
```

• We have to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each. The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius. Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the area of rectangle, square and circle respectively. Create an object of class 'Area' and call all the three methods.

Main.java

```java
    void SquareArea(int side) {
        int area = side * side;
        System.out.println("Area of square: " + area);
    }

    // Implement circle area calculation
    void CircleArea(double radius) {
        double area = Math.PI * radius * radius;
        System.out.println("Area of circle: " + area);
    }
}

public class TestShape {
    public static void main(String[] args) {
        Area area = new Area();

        area.RectangleArea(5, 10); // Print area of rectangle
        area.SquareArea(5); // Print area of square
        area.CircleArea(7.5); // Print area of circle
    }
}
```

Output

```
java -cp /tmp/dgW45vVCN8/TestShape
Area of rectangle: 50
Area of square: 25
Area of circle: 176.71458676442586

=== Code Execution Successful ===
```

I3. Write a program which will ask the user to enter his/her marks (out of 100). Define a method that will display grades according to the marks entered as below: Marks Grade 1-100 A 1-90 B 1-80 B 1-70 C 1-60 D 41-50 DD40 Fail

Main.java

```java
import java.util.Scanner;

public class DisplayGrades {
    // Methd to display grade based on marks
    static void displayGrade(int marks) {
        if (marks >= 91 && marks <= 100) {
            System.out.println("Grade: A");
        } else if (marks >= 81 && marks <= 90) {
            System.out.println("Grade: B");
        } else if (marks >= 71 && marks <= 80) {
            System.out.println("Grade: B");
        } else if (marks >= 61 && marks <= 70) {
            System.out.println("Grade: C");
        } else if (marks >= 51 && marks <= 60) {
            System.out.println("Grade: D");
        } else if (marks >= 41 && marks <= 50) {
            System.out.println("Grade: DD");
        } else if (marks <= 40) {
            System.out.println("Grade: Fail");
        } else {
            System.out.println("Invalid marks");
        }
```

Output

```
java -cp /tmp/eZDUO9TQco/DisplayGrades
Enter your marks (out of 100): 59
Grade: D

=== Code Execution Successful ===
```

• Create a class named 'Shape' with a method to print "This is this is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.

Main.java

```java
class Shape {
    // Methd to print shape
    void printShape() {
        System.out.println("This is this is shape");
    }
}

class Rectangle extends Shape {
    // Methd to print rectangle shape
    void printRectangle() {
        System.out.println("This is rectangular shape");
    }
}

class Circle extends Shape {
    // Methd to print circular shape
    void printCircle() {
        System.out.println("This is circular shape");
    }
}
```

Output

```
java -cp /tmp/qfDsqbYdwS/TestShape
This is this is shape
This is rectangular shape
Square is a rectangle

=== Code Execution Successful ===
```

• W.A.J. P to demonstrate try catch block,

Main.java

```java
public class TryCatchDemo {
    public static void main(String[] args) {
        try {
            int[] numbers = {1, 2, 3};
            System.out.println(numbers[3]); // Accessing invalid index
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Array index is out of bounds!");
        }
    }
}
```

Output

```
java -cp /tmp/C5uLNuhC1J/TryCatchDemo
Array index is out of bounds!

=== Code Execution Successful ===
```

• Take two numbers from the user and perform the division operation and handle Arithmetic Exception. O/P- Enter two numbers: 10 0 Exception in thread main java.lang.ArithmeticException:/ by zero

Main.java

```java
import java.util.Scanner;

public class DivisionDemo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter two numbers: ");
        int num1 = scanner.nextInt();
        int num2 = scanner.nextInt();

        try {
            int result = num1 / num2; // Division operation
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Exception in thread main java.lang
                .ArithmeticException: / by zero");
        }

        scanner.close();
    }
}
```

Output

```
java -cp /tmp/vlM4VJhkPd/DivisionDemo
Enter two numbers: 10 20
Result: 0

=== Code Execution Successful ===
```

• W.A.J. P to demonstrate multiple catch blocks, (one is to handle divide by zero exception and another one is to handle ArrayIndexOutOfBoundException) int a [] =new int [5]; a [5]=30/0;

Main.java

```java
public class MultipleCatchDemo {
    public static void main(String[] args) {
        try {
            int[] a = new int[5];
            a[5] = 30 / 0; // This will throw ArithmeticException
        } catch (ArithmeticException e) {
            System.out.println("ArithmeticException: " + e.getMessage
                ());
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("ArrayIndexOutOfBoundsException: " + e
                .getMessage());
        }
    }
}
```

Output

```
java -cp /tmp/ai1vpMSvB2/MultipleCatchDemo
ArithmeticException: / by zero

=== Code Execution Successful ===
```

• W.A.J. P to implement the above program (pro.no-B27) using nesting of try-catch block. try { try {//code} catch (Exception e) {//code} catch (Exception e) {//code}

Main.java

```java
public class NestedTryCatchDemo {
    public static void main(String[] args) {
        try {
            try {
                int result = 30 / 0; // This will throw ArithmeticException
            } catch (ArithmeticException e) {
                System.out.println("Inner catch: ArithmeticException: " + e.getMessage());
            }

            int[] a = new int[5];
            a[5] = 10; // This will throw ArrayIndexOutOfBoundsException
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Outer catch: ArrayIndexOutOfBoundsException: " + e.getMessage());
        }
    }
}
```

Output

```
java -cp /tmp/h2BH9d7E9S/NestedTryCatchDemo
Inner catch: ArithmeticException: / by zero
Outer catch: ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5

=== Code Execution Successful ===
```

• W.A.J. P to demonstrate try catch block, take two numbers from the user by Command line argument and perform the division operation and handle Arithmetic O/PException in thread main java. Lang. Arithmetic Exception:/ by zero

Main.java

```java
public class DivisionCommandLine {
    public static void main(String[] args) {
        if (args.length < 2) {
            System.out.println("Please provide two numbers as command line arguments.");
            return;
        }

        try {
            int num1 = Integer.parseInt(args[0]);
            int num2 = Integer.parseInt(args[1]);

            int result = num1 / num2; // Division operation
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Exception in thread main java.lang.ArithmeticException: / by zero");
        }
    }
}
```

Output

```
java -cp /tmp/dJeq0NXAEw/DivisionCommandLine
Please provide two numbers as command line arguments.

=== Code Execution Successful ===
```

• W.A.J.P to create the validate method that takes integer value as a parameter. If the age is less than 18, then throw an Arithmetic Exception otherwise print a message welcome to vote. O/P- Enter your age: 16 Exception in thread main java. Lang. Arithmetic Exception: not valid

Main.java

```java
import java.util.Scanner;

public class ValidateAge {
    // Methd to validate age
    static void validate(int age) {
        if (age < 18) {
            throw new ArithmeticException("not valid");
        } else {
            System.out.println("Welcome to vote");
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter your age: ");
        int age = scanner.nextInt();

        try {
            validate(age); // Validate age
        } catch (ArithmeticException e) {
```

Output

```
java -cp /tmp/R6Y7qN8B9R/ValidateAge
Enter your age: 12
ERROR!
Exception in thread main java.lang.ArithmeticException: not valid

=== Code Execution Successful ===
```

• W.A.J.P to create a custom exception if Customer withdraw amount which is greater than account balance then program will show custom exception otherwise amount will deduct from account balance. Account balance is: 2000 Enter withdraw amount: 2500

Main.java

```java
class InsufficientBalanceException extends Exception {
    // Constructor for custom exception
    InsufficientBalanceException(String message) {
        super(message);
    }
}

class Account {
    private int balance;

    // Constructor to initialize balance
    Account(int balance) {
        this.balance = balance;
    }

    // Method to withdraw amount
    void withdraw(int amount) throws InsufficientBalanceException {
        if (amount > balance) {
            int shortfall = amount - balance;
            throw new InsufficientBalanceException("Sorry,
                insufficient balance, you need more " + shortfall +
```

Output

```
java -cp /tmp/aQZyCxoAwn/TestAccount
Sorry, insufficient balance, you need more 500 Rs. To perform this
    transaction.

=== Code Execution Successful ===
```

Sorry, insufficient balance, you need more 500 Rs. To perform this transaction.

• W.A.J.P to create a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user defined exception "AgeNotWithinRangeException". If name contains numbers or special symbols raise exception "NameNotValidException". Define the two exception classes.

Main.java

```java
class AgeNotWithinRangeException extends Exception {
    // Constructor for age exception
    AgeNotWithinRangeException(String message) {
        super(message);
    }
}

class NameNotValidException extends Exception {
    // Constructor for name exception
    NameNotValidException(String message) {
        super(message);
    }
}

class Student {
    private int rollNo;
    private String name;
    private int age;
    private String course;

    // Parameterized constructor
```

Output

```
java -cp /tmp/obpxMrP5PQ/TestStudent
Age not within range (15-21)

=== Code Execution Successful ===
```

• W.A.J. P to create one thread by implementing Runnable interface in Class.

Main.java

```java
class MyRunnable implements Runnable {
    // Run method to be executed by thread
    public void run() {
        System.out.println("Thread running using Runnable interface."
            );
    }
}

public class TestRunnable {
    public static void main(String[] args) {
        Thread thread = new Thread(new MyRunnable());
        thread.start(); // Start the thread
    }
}
```

Output

```
java -cp /tmp/tyOUJxrdzq/TestRunnable
Thread running using Runnable interface.

=== Code Execution Successful ===
```

• W.A.J. P to create one thread by extending Thread class in another Class.

Main.java

```java
class MyThread extends Thread {
    // Override run method
    public void run() {
        System.out.println("Thread running by extending Thread class."
            );
    }
}

public class TestThread {
    public static void main(String[] args) {
        MyThread thread = new MyThread();
        thread.start(); // Start the thread
    }
}
```

Output

```
java -cp /tmp/ggSrORbvYP/TestThread
Thread running by extending Thread class.

=== Code Execution Successful ===
```

• W.A.J.P to create 2 threads and execute that threads by providing sleep time as 2000ms and check the execution.

Main.java

```java
class SleepThread extends Thread {
    public void run() {
        try {
            System.out.println("Thread started: " + Thread
                .currentThread().getId());
            Thread.sleep(2000); // Sleep for 2000 ms
            System.out.println("Thread ended: " + Thread
                .currentThread().getId());
        } catch (InterruptedException e) {
            System.out.println("Thread interrupted.");
        }
    }
}

public class TestSleepThreads {
    public static void main(String[] args) {
        SleepThread t1 = new SleepThread();
        SleepThread t2 = new SleepThread();

        t1.start(); // Start first thread
        t2.start(); // Start second thread
    }
```

Output

```
java -cp /tmp/pT5y7Gtxv7/TestSleepThreads
Thread started: 11
Thread started: 10
Thread ended: 11
Thread ended: 10

=== Code Execution Successful ===
```

• W.A.J.P to start the same Thread twice by calling start () method twice. Test ThreadTwice1 t1=new TestThreadTwice1(); t1.start (); t1.start ();

Main.java

```java
class TestThreadTwice1 extends Thread {
    public void run() {
        System.out.println("Thread running.");
    }
}

public class TestThreadTwice {
    public static void main(String[] args) {
        TestThreadTwice1 t1 = new TestThreadTwice1();
        t1.start(); // Start thread first time
        try {
            t1.start(); // Attempt to start thread second time
        } catch (IllegalThreadStateException e) {
            System.out.println("Thread already started and cannot be
                restarted.");
        }
    }
}
```

Output

```
java -cp /tmp/JxCyFEMJfX/TestThreadTwice
Thread already started and cannot be restarted.
Thread running.

=== Code Execution Successful ===
```

• W.A.J.P to create 2 threads and make one thread as Daemon Thread by using set Daemon () method of Thread class and check whether the thread is set daemon or not by using is Daemon () method.
TestDaemonThread2 t1=new TestDaemonThread2();
TestDaemonThread2 t2=new TestDaemonThread2(); t1.start();
t1.setDaemon(true);//will throw exception here t2.start();

Main.java

```java
class TestDaemonThread2 extends Thread {
    public void run() {
        System.out.println("Daemon thread running.");
    }
}

public class TestDaemon {
    public static void main(String[] args) {
        TestDaemonThread2 t1 = new TestDaemonThread2();
        TestDaemonThread2 t2 = new TestDaemonThread2();

        t1.setDaemon(true); // Set t1 as daemon thread
        t1.start(); // Start daemon thread

        t2.start(); // Start normal thread

        System.out.println("Is t1 a daemon thread? " + t1.isDaemon());
    }
}
```

Output

```
java -cp /tmp/ZlfwcAtJPB/TestDaemon
Daemon thread running.
Daemon thread running.
Is t1 a daemon thread? true

=== Code Execution Successful ===
```

Write a Java program to create a new array list, add some colors (string) and print out the collection.

Main.java

```java
import java.util.ArrayList;
import java.util.Collections;

public class ArrayListDemo {
    public static void main(String[] args) {
        // Create a new ArrayList
        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");

        // Print all elements
        System.out.println("ArrayList elements: " + colors);

        // Iterate through all elements
        for (String color : colors) {
            System.out.println(color);
        }

        // Insert element at first position
        colors.add(0, "Pink");
```

Output

```
java -cp /tmp/6G5xvEI8U8/ArrayListDemo
ArrayList elements: [Red, Green, Blue, Yellow]
Red
Green
Blue
Yellow
After insertion: [Pink, Red, Green, Blue, Yellow]
Element at index 2: Green
After update: [Pink, Red, Purple, Blue, Yellow]
After removal: [Pink, Red, Blue, Yellow]
Contains Green? false
Sorted list: [Blue, Pink, Red, Yellow]
Copied list: [Blue, Pink, Red, Yellow]
Shuffled list: [Yellow, Red, Pink, Blue]

=== Code Execution Successful ===
```

• Write a Java program to iterate through all elements in an array list.

Main.java

```java
import java.util.HashSet;

public class HashSetDemo {
    public static void main(String[] args) {
        HashSet<String> set = new HashSet<>();
        set.add("Red");
        set.add("Green");
        set.add("Blue");

        // Append element to the set
        set.add("Yellow");
        System.out.println("HashSet elements: " + set);
    }
}
```

Output

```
java -cp /tmp/JSKKtvumiu/HashSetDemo
HashSet elements: [Red, Blue, Yellow, Green]

=== Code Execution Successful ===
```

• Write a Java program to insert an element into the array list at the first position.

Main.java

```java
import java.util.ArrayList;

public class InsertElement {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("Second");
        list.add("Third");

        // Insert element at the first position
        list.add(0, "First");

        System.out.println("ArrayList after insertion: " + list);
    }
}
```

Output

```
java -cp /tmp/kI1deSjxhU/InsertElement
ArrayList after insertion: [First, Second, Third]

=== Code Execution Successful ===
```

• Write a Java program to retrieve an element (at a specified index) from a given array List.

Main.java

```java
import java.util.ArrayList;

public class RetrieveElement {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("First");
        list.add("Second");
        list.add("Third");

        // Retrieve element at index 1
        String element = list.get(1);

        System.out.println("Element at index 1: " + element);
    }
}
```

Output

```
java -cp /tmp/69I7YIr6wT/RetrieveElement
Element at index 1: Second

=== Code Execution Successful ===
```

• Write a Java program to update specific array element by given element.

Main.java

```java
import java.util.ArrayList;

public class UpdateElement {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("First");
        list.add("Second");
        list.add("Third");

        // Update element at index 1
        list.set(1, "Updated Second");

        System.out.println("ArrayList after update: " + list);
    }
}
```

Output

```
java -cp /tmp/RF9cjB5ghZ/UpdateElement
ArrayList after update: [First, Updated Second, Third]

=== Code Execution Successful ===
```

• Write a Java program to remove the third element from an array list.

Main.java

```java
import java.util.ArrayList;

public class RemoveElement {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("First");
        list.add("Second");
        list.add("Third");
        list.add("Fourth");

        // Remove the third element
        list.remove(2);

        System.out.println("ArrayList after removal: " + list);
    }
}
```

Output

```
java -cp /tmp/T1vOqeDBj3/RemoveElement
ArrayList after removal: [First, Second, Fourth]

=== Code Execution Successful ===
```

• Write a Java program to search an element in an array list.

Main.java

```java
import java.util.ArrayList;

public class SearchElement {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("First");
        list.add("Second");
        list.add("Third");

        // Search for an element
        boolean exists = list.contains("Second");

        System.out.println("Contains 'Second'? " + exists);
    }
}
```

Output

```
java -cp /tmp/x6O1GInOA6/SearchElement
Contains 'Second'? true

=== Code Execution Successful ===
```

• Write a Java program to sort a given array list.

Main.java

```java
import java.util.ArrayList;
import java.util.Collections;

public class SortArrayList {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("Banana");
        list.add("Apple");
        list.add("Cherry");

        // Sort the ArrayList
        Collections.sort(list);

        System.out.println("Sorted ArrayList: " + list);
    }
}
```

Output

```
java -cp /tmp/nUlWDerLpF/SortArrayList
Sorted ArrayList: [Apple, Banana, Cherry]

=== Code Execution Successful ===
```

• Write a Java program to copy one array list into another.

Main.java

```java
import java.util.ArrayList;

public class CopyArrayList {
    public static void main(String[] args) {
        ArrayList<String> originalList = new ArrayList<>();
        originalList.add("First");
        originalList.add("Second");
        originalList.add("Third");

        // Copy originalList to newList
        ArrayList<String> newList = new ArrayList<>(originalList);

        System.out.println("Copied ArrayList: " + newList);
    }
}
```

Output

```
java -cp /tmp/f4HeTuPfST/CopyArrayList
Copied ArrayList: [First, Second, Third]

=== Code Execution Successful ===
```

• Write a Java program to shuffle elements in an array list.

Main.java

```java
import java.util.ArrayList;
import java.util.Collections;

public class ShuffleArrayList {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("First");
        list.add("Second");
        list.add("Third");

        // Shuffle the ArrayList
        Collections.shuffle(list);

        System.out.println("Shuffled ArrayList: " + list);
    }
}
```

Output

```
java -cp /tmp/lQwjyFRLA4/ShuffleArrayList
Shuffled ArrayList: [Third, Second, First]

=== Code Execution Successful ===
```

• Write a Java program to append the specified element to the end of a hash set

Main.java

```java
import java.util.HashSet;

public class AppendToHashSet {
    public static void main(String[] args) {
        HashSet<String> set = new HashSet<>();
        set.add("Red");
        set.add("Green");

        // Append an element to the end of the HashSet
        set.add("Blue");

        System.out.println("HashSet after appending: " + set);
    }
}
```

Output

```
java -cp /tmp/MCRliWVPk2/AppendToHashSet
HashSet after appending: [Red, Blue, Green]

=== Code Execution Successful ===
```

• Write a Java program to iterate through all elements in a hash list.

Main.java

```java
import java.util.HashSet;

public class IterateHashSet {
    public static void main(String[] args) {
        HashSet<String> set = new HashSet<>();
        set.add("Red");
        set.add("Green");
        set.add("Blue");

        // Iterate through HashSet
        for (String color : set) {
            System.out.println(color);
        }
    }
}
```

Output

```
java -cp /tmp/d0klS7uJSN/IterateHashSet
Red
Blue
Green

=== Code Execution Successful ===
```

• Write a Java program to get the number of elements in a hash set.

Main.java

```java
import java.util.HashSet;

public class CountHashSetElements {
    public static void main(String[] args) {
        HashSet<String> set = new HashSet<>();
        set.add("Red");
        set.add("Green");
        set.add("Blue");

        // Get number of elements
        int size = set.size();

        System.out.println("Number of elements in HashSet: " + size);
    }
}
```

Output

```
java -cp /tmp/ZBu5FsRJOc/CountHashSetElements
Number of elements in HashSet: 3

=== Code Execution Successful ===
```

• Write a Java program to associate the specified value with the specified key in a Hash Map.

Main.java

```java
import java.util.HashMap;

public class PutInHashMap {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();

        // Associate value with key
        map.put("Alice", 25);
        map.put("Bob", 30);

        System.out.println("HashMap: " + map);
    }
}
```

Output

```
java -cp /tmp/xW4wYBfwI2/PutInHashMap
HashMap: {Bob=30, Alice=25}

=== Code Execution Successful ===
```

• Write a Java program to count the number of key-value (size) mappings in a map.

Main.java

```java
import java.util.HashMap;

public class CountMapEntries {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("Alice", 25);
        map.put("Bob", 30);

        // Count number of key-value mappings
        int size = map.size();

        System.out.println("Number of key-value mappings in HashMap: "
            + size);
    }
}
```

Output

```
java -cp /tmp/RJRvAJaumg/CountMapEntries
Number of key-value mappings in HashMap: 2

=== Code Execution Successful ===
```

• Write a Java program to reverse elements in an array list.

Main.java

```java
import java.util.ArrayList;
import java.util.Collections;

public class ReverseArrayList {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("First");
        list.add("Second");
        list.add("Third");

        // Reverse the ArrayList
        Collections.reverse(list);

        System.out.println("Reversed ArrayList: " + list);
    }
}
```

Output

```
java -cp /tmp/VGDt6mqTPi/ReverseArrayList
Reversed ArrayList: [Third, Second, First]

=== Code Execution Successful ===
```

• Write a Java program to extract a portion of an array list.

Main.java

```java
import java.util.ArrayList;

public class SubListArrayList {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("First");
        list.add("Second");
        list.add("Third");
        list.add("Fourth");

        // Extract a portion of the ArrayList
        ArrayList<String> subList = new ArrayList<>(list.subList(1, 3
            ));

        System.out.println("SubList: " + subList);
    }
}
```

Output

```
java -cp /tmp/qhZvpwOEL9/SubListArrayList
SubList: [Second, Third]

=== Code Execution Successful ===
```

• Write a Java program to compare two array lists.

```java
import java.util.ArrayList;

public class CompareArrayLists {
    public static void main(String[] args) {
        ArrayList<String> list1 = new ArrayList<>();
        list1.add("A");
        list1.add("B");

        ArrayList<String> list2 = new ArrayList<>();
        list2.add("A");
        list2.add("B");

        // Compare ArrayLists
        boolean isEqual = list1.equals(list2);

        System.out.println("Are the two ArrayLists equal? " + isEqual
            );
    }
}
```

```
java -cp /tmp/EC1riUQJq9/CompareArrayLists
Are the two ArrayLists equal? true

=== Code Execution Successful ===
```

• Write a Java program of swap two elements in an array list.

```java
import java.util.ArrayList;

public class SwapArrayListElements {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("First");
        list.add("Second");
        list.add("Third");

        // Swap elements at index 0 and 2
        String temp = list.get(0);
        list.set(0, list.get(2));
        list.set(2, temp);

        System.out.println("ArrayList after swap: " + list);
    }
}
```

```
java -cp /tmp/cHWNyj5WRR/SwapArrayListElements
ArrayList after swap: [Third, Second, First]

=== Code Execution Successful ===
```

• Write a Java program to join two array lists.

Main.java

```java
import java.util.ArrayList;

public class JoinArrayLists {
    public static void main(String[] args) {
        ArrayList<String> list1 = new ArrayList<>();
        list1.add("A");
        list1.add("B");

        ArrayList<String> list2 = new ArrayList<>();
        list2.add("C");
        list2.add("D");

        // Join two ArrayLists
        list1.addAll(list2);

        System.out.println("Joined ArrayList: " + list1);
    }
}
```

Output

```
java -cp /tmp/vRBcuDNifa/JoinArrayLists
Joined ArrayList: [A, B, C, D]

=== Code Execution Successful ===
```

• Write a Java program to convert a hash set to an array.

Main.java

```java
import java.util.HashSet;

public class HashSetToArray {
    public static void main(String[] args) {
        HashSet<String> set = new HashSet<>();
        set.add("Red");
        set.add("Green");
        set.add("Blue");

        // Convert HashSet to array
        String[] array = set.toArray(new String[0]);

        System.out.println("HashSet converted to array:");
        for (String color : array) {
            System.out.println(color);
        }
    }
}
```

Output

```
java -cp /tmp/G21RfRqEZJ/HashSetToArray
HashSet converted to array:
Red
Blue
Green

=== Code Execution Successful ===
```

• Write a Java program to convert a hash set to a List/Array List.

Main.java

```java
import java.util.ArrayList;
import java.util.HashSet;

public class HashSetToArrayList {
    public static void main(String[] args) {
        HashSet<String> set = new HashSet<>();
        set.add("Red");
        set.add("Green");
        set.add("Blue");

        // Convert HashSet to ArrayList
        ArrayList<String> list = new ArrayList<>(set);

        System.out.println("HashSet converted to ArrayList: " + list);
    }
}
```

Output

```
java -cp /tmp/9zhm8bb3E7/HashSetToArrayList
HashSet converted to ArrayList: [Red, Blue, Green]

=== Code Execution Successful ===
```

• Write a Java program to check whether a map contains key-value mappings (empty) or not.

Main.java

```java
import java.util.HashMap;

public class CheckMapEmpty {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();

        // Check if the map is empty
        boolean isEmpty = map.isEmpty();

        System.out.println("Is the map empty? " + isEmpty);
    }
}
```

Output

```
java -cp /tmp/KQSGWTqUhH/CheckMapEmpty
Is the map empty? true

=== Code Execution Successful ===
```

• Write a Java program to increase the size of an array list.

```java
import java.util.ArrayList;

public class IncreaseArrayListSize {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("Element 1");
        list.add("Element 2");

        // Increase the size of the ArrayList
        list.ensureCapacity(10);  // Ensure capacity of at least 10

        System.out.println("ArrayList capacity increased");
    }
}
```

```
java -cp /tmp/Yw6zWBZqcP/IncreaseArray
ArrayList capacity increased

=== Code Execution Successful ===
```

• Write a Java program to replace the second element of an Array List with the specified element.

Main.java

```java
import java.util.ArrayList;

public class ReplaceElementInArrayList {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("First");
        list.add("Second");
        list.add("Third");

        // Replace the second element
        list.set(1, "Updated Second");

        System.out.println("ArrayList after replacing second element:
            " + list);
    }
}
```

Output

```
java -cp /tmp/2TjKFuFzTG/ReplaceElementInArrayList
ArrayList after replacing second element: [First, Updated Second, Third]

=== Code Execution Successful ===
```

• Write a Java program to print all the elements of an Array List using the position of the elements.

```java
import java.util.ArrayList;

public class PrintElementsByPosition {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("First");
        list.add("Second");
        list.add("Third");

        // Print all elements by position
        for (int i = 0; i < list.size(); i++) {
            System.out.println("Element at position " + i + ": " +
                list.get(i));
        }
    }
}
```

```
java -cp /tmp/2Ys46G0Oni/PrintElementsByPosit
Element at position 0: First
Element at position 1: Second
Element at position 2: Third

=== Code Execution Successful ===
```

• Write a Java program to compare two sets and retain elements which are same on both sets.

Main.java Share Run Output

```java
import java.util.HashSet;

public class CompareSets {
    public static void main(String[] args) {
        HashSet<String> set1 = new HashSet<>();
        set1.add("A");
        set1.add("B");
        set1.add("C");

        HashSet<String> set2 = new HashSet<>();
        set2.add("B");
        set2.add("C");
        set2.add("D");

        // Retain common elements
        set1.retainAll(set2);

        System.out.println("Common elements: " + set1);
    }
}
```

```
java -cp /tmp/IDs3yaTvIq/CompareSets
Common elements: [B, C]

=== Code Execution Successful ===
```

• Write a Java program to get a collection view of the values contained in this map.

Main.java

Share Run

```java
import java.util.HashMap;
import java.util.Collection;

public class MapValuesCollection {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("Alice", 25);
        map.put("Bob", 30);

        // Get a collection view of values
        Collection<Integer> values = map.values();

        System.out.println("Values in the map: " + values);
    }
}
```

Output

```
java -cp /tmp/FgtrIlhcDw/MapValuesCollectio
Values in the map: [30, 25]

=== Code Execution Successful ===
```