MANAV RACHNA INTERNATIONAL

INSTITUTE OF RESEARCH AND STUDIES

FACULTY OF COMPUTER APPLICATIONS

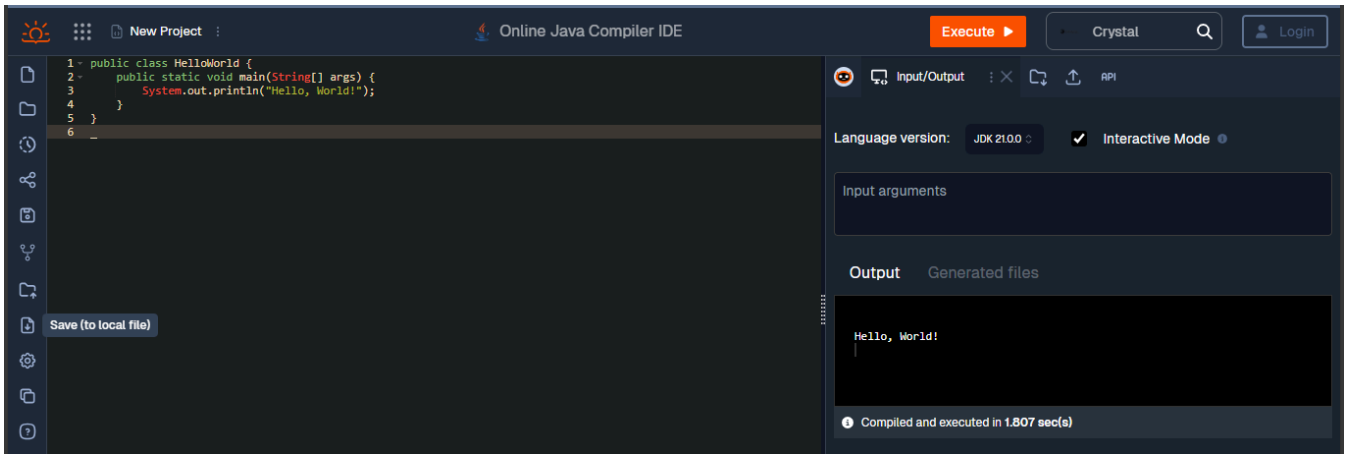
PRACTICAL FILE

OF

JAVA LAB (OBCA-DS-454)

Submitted By	
Student Name	**DISHA THAKRAN**
Roll No	**23/SCA/BCA(DS & BDA)/012**
Programme	**Bachelor of Computer Applications**
Semester	**4th**
Section/Group	**E**
Department	**Computer Applications**
Batch	**2023-26**
Submitted To	
Faculty Name	**Ms. Priyanka Sharma**

Q1. Write java program to print hello world

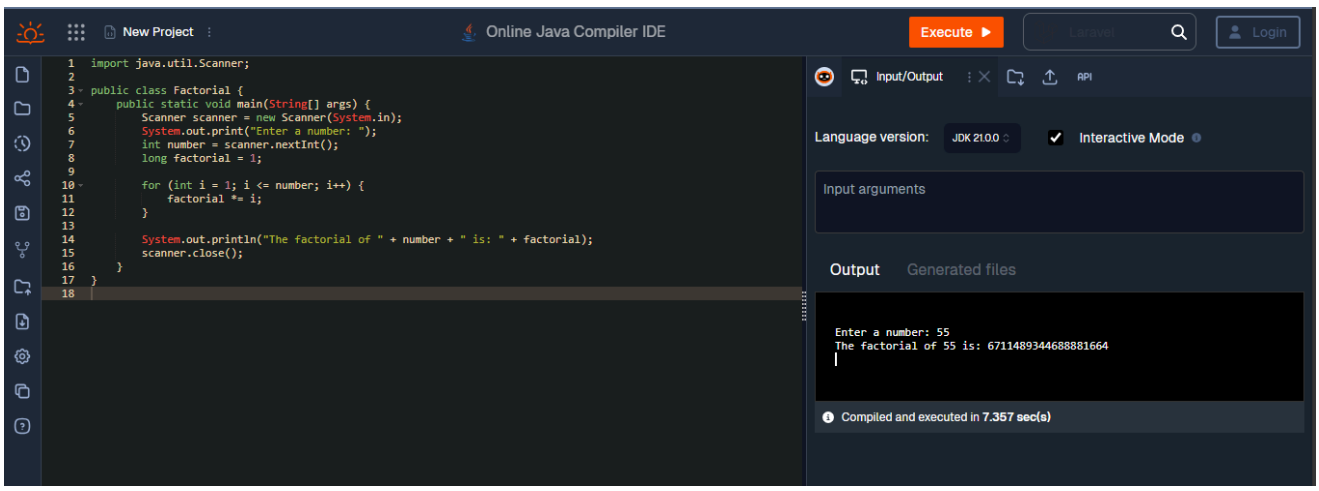


Q2. Java Program to take input from user and print the sum of two numbers

Q3. Create a java program to check whether a number entered by user is even or odd

Q4.Create a java program to print the average and sum of 5 numbers entered by user.

Q5. Program to calculate the factorial of a number



Q6. Program to print Fibonacci series up to n terms.

Q7.Program to reverse a number

Main.java

```java

import java.util.Scanner;
 public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        int reversedNumber = 0;

        while(number != 0){
            int digit= number %10;
            reversedNumber= reversedNumber * 10 + digit;
            number = number/10;

        }
        System.out.println("Reversed Number: " + reversedNumber);

    }
}
```

Output

```
Enter a number: 23456
Reversed Number: 65432

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

Q8.Program to check if a number is a palindrome

Main.java

```java

public class Main {
    public static void main(String[] args) {
        int number = 121;
        if (isPalindrome(number)) {
            System.out.println(number + " is a palindrome.");
        } else {
            System.out.println(number + " is not a palindrome.");
        }
    }

    public static boolean isPalindrome(int number) {
        int originalNumber = number;
        int reversedNumber = 0;
        while (number != 0) {
            int digit = number % 10;
            reversedNumber = reversedNumber * 10 + digit;
            number /= 10;
        }
        return originalNumber == reversedNumber;
    }
}
```

Output

```
121 is a palindrome.

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

Q9.Program for a simple calculator

Main.java

```java
import java.util.Scanner;

class Main {
  public static void main(String[] args) {
    char operator;
    Double number1, number2, result;
    Scanner input = new Scanner(System.in);
    System.out.println("Choose an operator: +, -, *, or /");
    operator = input.next().charAt(0);
    System.out.println("Enter first number");
    number1 = input.nextDouble();
    System.out.println("Enter second number");
    number2 = input.nextDouble();
    switch (operator) {
      case '+':
        result = number1 + number2;
        System.out.println(number1 + " + " + number2 + " = " + result);
        break;
      case '-':
        result = number1 - number2;
        System.out.println(number1 + " - " + number2 + " = " + result);
        break;
      case '*':
        result = number1 * number2;
        System.out.println(number1 + " * " + number2 + " = " + result);
        break;
      case '/':
        result = number1 / number2;
        System.out.println(number1 + " / " + number2 + " = " + result);
        break;
      default:
        System.out.println("Invalid operator!");
        break;
    }
    input.close();
  }
}
```

Output

```
Choose an operator: +, -, *, or /
*
Enter first number
5
Enter second number
2
5.0 * 2.0 = 10.0

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

Q10.Program to check if a number is prime

Main.java

```java
public class Main {
    public static void main(String[] args) {
        int number = 29;
        boolean isPrime = checkPrime(number);

        if (isPrime) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }
    }
    public static boolean checkPrime(int num) {
        if (num <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

Output

```
29 is a prime number.

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

Q11. Program to check if a number is an Armstrong number

Online Java Compiler IDE

```java
import java.util.Scanner;

public class ArmstrongNumberCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        int originalNumber = number;
        int result = 0;
        int numberOfDigits = String.valueOf(number).length();

        while (number != 0) {
            int digit = number % 10;
            result += Math.pow(digit, numberOfDigits);
            number /= 10;
        }

        if (result == originalNumber) {
            System.out.println(originalNumber + " is an Armstrong number.");
        } else {
            System.out.println(originalNumber + " is not an Armstrong number.");
        }

        scanner.close();
    }
}
```

Input/Output

Language version: JDK 21.0.0

Interactive Mode

Input arguments

Output

```
Enter a number: 87
87 is not an Armstrong number.
```

Compiled and executed in 5.298 sec(s)

Q12.Find the Largest of Three Numbers using ternary operator

Online Java Compiler IDE

```java
import java.util.Scanner;

public class LargestNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the first number: ");
        int num1 = scanner.nextInt();
        System.out.print("Enter the second number: ");
        int num2 = scanner.nextInt();

        // Using the ternary operator to find the largest number
        int largest = (num1 > num2) ? num1 : num2;

        System.out.println("The largest number is: " + largest);
        scanner.close();
    }
}
```

Input/Output

Language version: JDK 21.0.0

Interactive Mode

Input arguments

Output

```
Enter the first number: 54
Enter the second number: 84
The largest number is: 84
```

Compiled and executed in 12.316 sec(s)

Q13.Print Multiplication Table

Main.java

```java
public class Main {

    public static void main(String[] args) {

        int num = 5;
        for(int i = 1; i <= 10; ++i)
        {
            System.out.printf("%d * %d = %d \n", num, i, num * i);
        }
    }
}
```

Output

```
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50

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

Q14. Calculate Sum and Average of Array Elements

Main.java

```java
import java.util.Scanner;
public class Main
{
    public static void main(String[] args)    {
        int n, sum = 0;
        float average;
        Scanner s = new Scanner(System.in);
        System.out.print("Enter no. of elements you want in array:"
            );
        n = s.nextInt();
        int a[] = new int[n];
        System.out.println("Enter all the elements:");
        for(int i = 0; i < n ; i++)
        {
            a[i] = s.nextInt();
            sum = sum + a[i];
        }
        System.out.println("Sum:"+sum);
        average = (float)sum / n;
        System.out.println("Average:"+average);
    }
}
```

Output

```
Enter no. of elements you want in array:
5
Enter all the elements:
2 5 7 8 9
Sum:31
Average:6.2

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

Q15. Reverse a String

Online Java Compiler IDE

```java
import java.util.Scanner;

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

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

        String reversed = new StringBuilder(input).reverse().toString();

        System.out.println("Reversed string: " + reversed);
    }
}
```

Input/Output

```
Enter a string: JAVA
Reversed string: AVAJ
```

Compiled and executed in 9.899 sec(s)

Q16.Find Factorial of a Number Using Recursion

Main.java

```java
public class Main {

    public static void main(String[] args) {
        int num = 6;
        long factorial = multiplyNumbers(num);
        System.out.println("Factorial of " + num + " = " + factorial);
    }
    public static long multiplyNumbers(int num)
    {
        if (num >= 1)
            return num * multiplyNumbers(num - 1);
        else
            return 1;
    }
}
```

Output

```
Factorial of 6 = 720

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

Q17. Sort an Array in Ascending Order

Main.java

```java
public class Main {
    public static void main(String[] args) {
        int [] arr = new int [] {5, 2, 8, 7, 1};
        int temp = 0;
        System.out.println("Elements of original array: ");
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        for (int i = 0; i < arr.length; i++) {
            for (int j = i+1; j < arr.length; j++) {
               if(arr[i] > arr[j]) {
                   temp = arr[i];
                   arr[i] = arr[j];
                   arr[j] = temp;
               }
            }       }
        System.out.println();
        System.out.println("Elements of array sorted in ascending order: ");

        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
    }
}
```

Output

```
Elements of original array:
5 2 8 7 1
Elements of array sorted in ascending order:
1 2 5 7 8
=== Code Execution Successful ===
```

Q18. Check Palindrome for a String

Online Java Compiler IDE

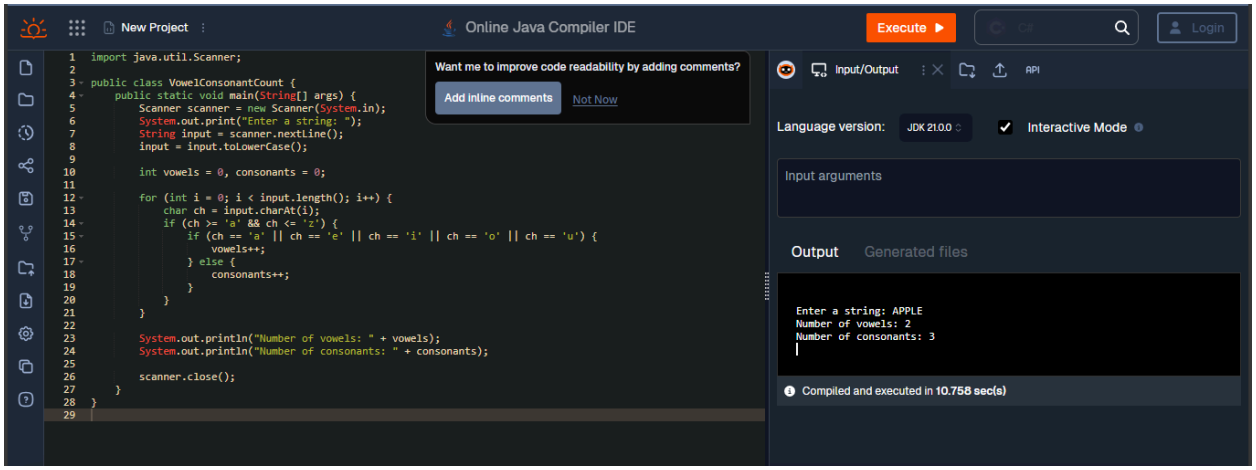
```java
import java.util.Scanner;
public class PalindromeCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();
        String reversedString = new StringBuilder(input).reverse().toString();

        if (input.equalsIgnoreCase(reversedString)) {
            System.out.println(input + " is a palindrome.");
        } else {
            System.out.println(input + " is not a palindrome.");
        }

        scanner.close();
    }
}
```

Input/Output

```
Enter a string: aashi
aashi is not a palindrome.
```

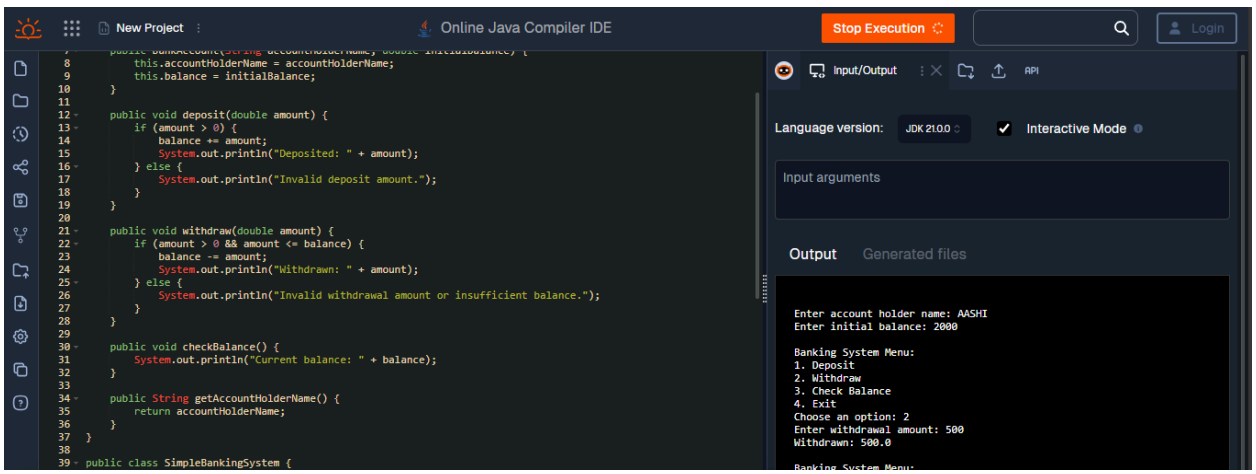
Compiled and executed in 7.267 sec(s)

Q19. Count Vowels and Consonants in a String



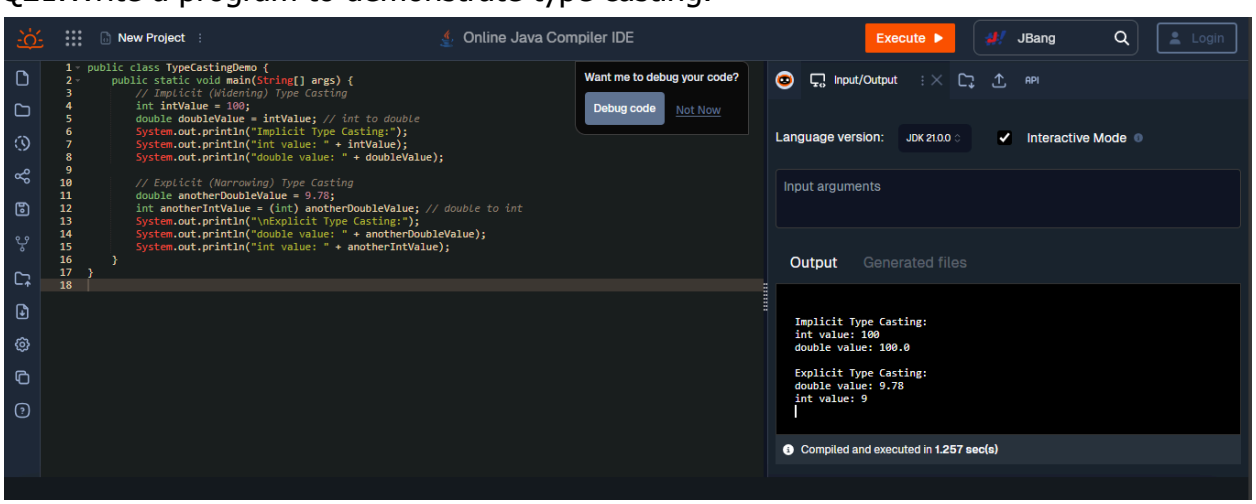

```java
import java.util.Scanner;

public class VowelConsonantCount {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();
        input = input.toLowerCase();

        int vowels = 0, consonants = 0;

        for (int i = 0; i < input.length(); i++) {
            char ch = input.charAt(i);
            if (ch >= 'a' && ch <= 'z') {
                if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
                    vowels++;
                } else {
                    consonants++;
                }
            }
        }

        System.out.println("Number of vowels: " + vowels);
        System.out.println("Number of consonants: " + consonants);

        scanner.close();
    }
}
```

Q20.Implement a Simple Banking System




```java
        this.accountHolderName = accountHolderName;
        this.balance = initialBalance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient balance.");
        }
    }

    public void checkBalance() {
        System.out.println("Current balance: " + balance);
    }

    public String getAccountHolderName() {
        return accountHolderName;
    }
}

public class SimpleBankingSystem {
```

Q21.Write a program to demonstrate type casting.

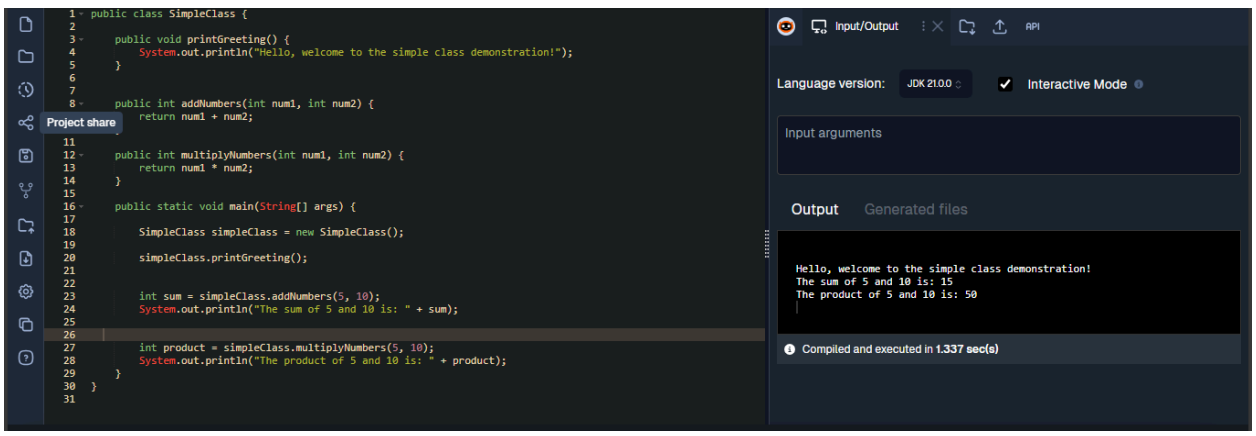



```java
public class TypeCastingDemo {
    public static void main(String[] args) {
        // Implicit (Widening) Type Casting
        int intValue = 100;
        double doubleValue = intValue; // int to double
        System.out.println("Implicit Type Casting:");
        System.out.println("int value: " + intValue);
        System.out.println("double value: " + doubleValue);

        // Explicit (Narrowing) Type Casting
        double anotherDoubleValue = 9.78;
        int anotherIntValue = (int) anotherDoubleValue; // double to int
        System.out.println("\nExplicit Type Casting:");
        System.out.println("double value: " + anotherDoubleValue);
        System.out.println("int value: " + anotherIntValue);
    }
}
```

Q22.Write a program to generate prime numbers between 1 & given number


```java
import java.util.Scanner;

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

        System.out.println("Prime numbers between 1 and " + n + " are:");
        for (int i = 2; i <= n; i++) {
            if (isPrime(i)) {
                System.out.print(i + " ");
            }
        }
        scanner.close();
    }

    public static boolean isPrime(int num) {
        if (num <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

Q23. Program to Demonstrate a Simple Class with Methods

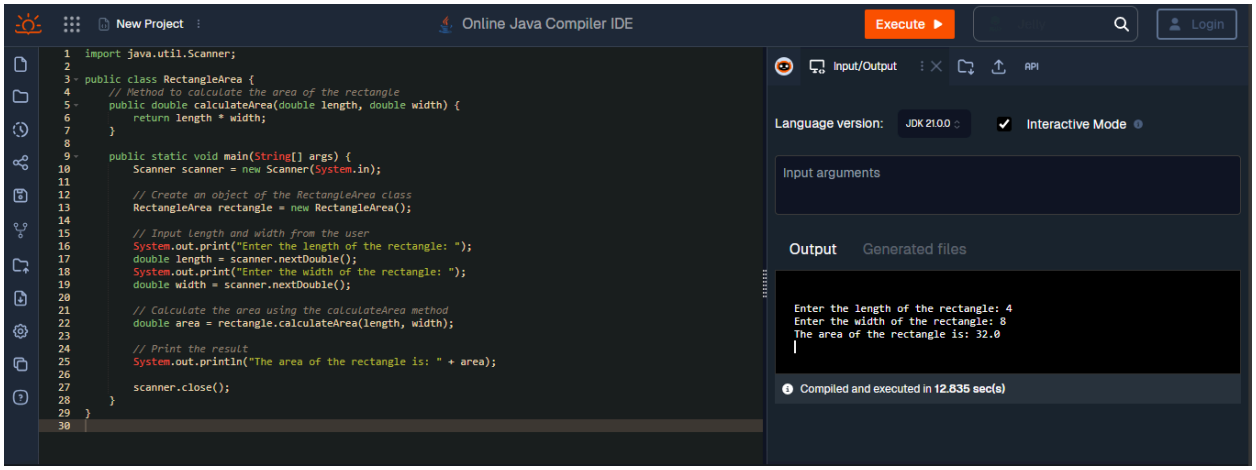



```java
public class SimpleClass {

    public void printGreeting() {
        System.out.println("Hello, welcome to the simple class demonstration!");
    }


    public int addNumbers(int num1, int num2) {
        return num1 + num2;


    public int multiplyNumbers(int num1, int num2) {
        return num1 * num2;
    }

    public static void main(String[] args) {

        SimpleClass simpleClass = new SimpleClass();

        simpleClass.printGreeting();


        int sum = simpleClass.addNumbers(5, 10);
        System.out.println("The sum of 5 and 10 is: " + sum);


        int product = simpleClass.multiplyNumbers(5, 10);
        System.out.println("The product of 5 and 10 is: " + product);
    }
}
```

Q24.Program for Class with Parameterized Constructor

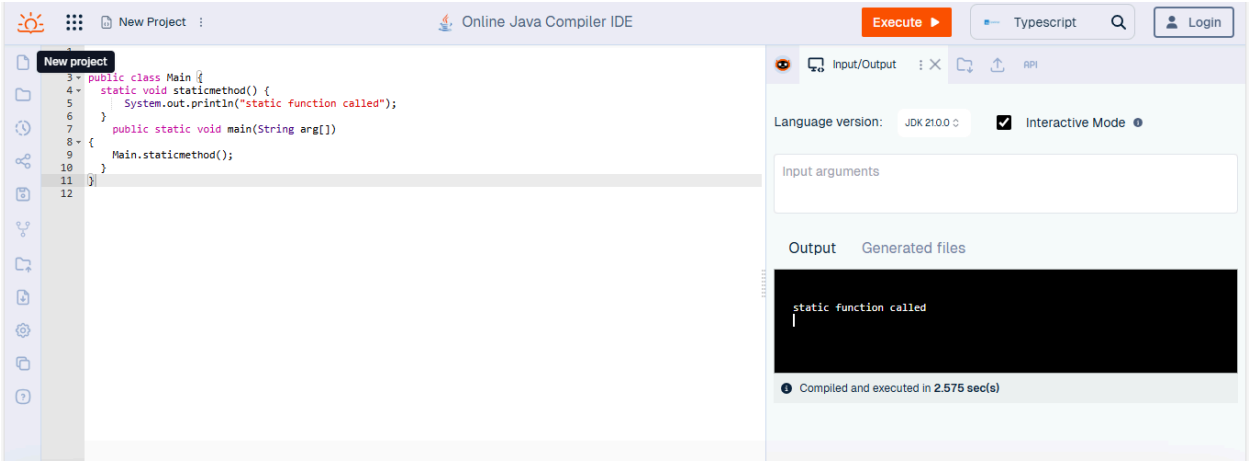

```java
public class Student {
    private String name;
    private int age;
    private String grade;

    // Parameterized constructor
    public Student(String name, int age, String grade) {
        this.name = name;
        this.age = age;
        this.grade = grade;
    }

    // Method to display student details
    public void displayDetails() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Grade: " + grade);
    }

    public static void main(String[] args) {
        // Create an object of the Student class using the parameterized constructor
        Student student = new Student("Alice", 20, "A");

        // Call the displayDetails method
        student.displayDetails();
    }
}
```

Q25.Program to Find the Area of a Rectangle Using Methods



Q26.Program for Bank Account Class with Deposit and Withdraw Methods

Q27. Program to Demonstrate Static Methods



Q28. Program to Demonstrate Method Overriding

Main.java

```java
class ABC
{
    public void method()
    {
        System.out.println("base call function");
    }
}
    public class Main extends ABC
    {
        public static void method1()
            {
                System.out.println("child class function");
            }
            public static void main(String args[])
            {
                Main xy=new Main();
                xy.method();
            }
        }
```

Output

```
base call function

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

Q29.Program to Demonstrate Getters and Setters

Main.java

```java
class Student {
  private String name;
  private int rollno;
  public void setName(String n) {
    this.name = n;
  }

  public String getName() {
    return name;
  }
  public void setRollno(int r) {
    this.rollno = r;
  }
  public int getRollno() {
    return rollno;
  }
}
public class Main {

  public static void main(String[] args) {
    Student s1 = new Student();

    s1.setName("Scaler");
    s1.setRollno(27);
    System.out.println(s1.getName());

    System.out.println(s1.getRollno());
  }
}
```

Output

```
Scaler
27

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

Q30.Program to Demonstrate a Class with Multiple Methods

Main.java

```java
class Main {

  public int addNumbers(int a, int b) {
    int sum = a + b;
    return sum;
  }

  public static void main(String[] args) {

    int num1 = 25;
    int num2 = 15;

    Main obj = new Main();
    int result = obj.addNumbers(num1, num2);
    System.out.println("Sum is: " + result);
  }
}
```

Output

```
Sum is: 40

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

Q31.Write a program to create a simple class to find out the area and perimeter of rectangle

using super and this keyword.

Main.java

```java
import java.util.Scanner;
public class Main
{
    public static void main(String[] args)
    {
        int l, b, perimeter, area;
        Scanner s = new Scanner(System.in);
        System.out.print("Enter length of rectangle:");
        l = s.nextInt();
        System.out.print("Enter breadth of rectangle:");
        b = s.nextInt();
        perimeter = 2 * (l + b);
        System.out.println("Perimeter of rectangle:"+perimeter);
        area = l * b;
        System.out.println("Area of rectangle:"+area);
    }
}
```

Output

```
Enter length of rectangle:8
Enter breadth of rectangle:10
Perimeter of rectangle:36
Area of rectangle:80

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

Q32.Write a program to count the number of objects created for a class using static member function

Number_objects.java

```java
public class Number_Objects
{
    static int count=0;
    Number_Objects()
    {
        count++;
    }
    public static void main(String[] args)
    {
        Number_Objects obj1 = new Number_Objects();
        Number_Objects obj2 = new Number_Objects();
        Number_Objects obj3 = new Number_Objects();
        Number_Objects obj4 = new Number_Objects();
        System.out.println("Number of objects created:"+count);
    }
}
```

STDIN

Input for the program (Optional)

Output:

```
Number of objects created:4
```

Q33. Write a program to design a class using abstract methods and abstract classes.

```java
abstract class Shape {
    abstract double area();

    void display() {
        System.out.println("This is shape");
    }
}

class Circle extends Shape {
    private double radius;

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

    @Override
    double area() {
        return Math.PI * radius * radius;
    }
}

class Rectangle extends Shape {
    private double length, width;

    Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
}
    @Override
    double area() {
        return length * width;
    }
}

class Main {
    public static void main(String[] args) {
        Circle circle = new Circle(3);
        Shape rect = new Rectangle(4, 6);

        System.out.println("Circle area: " + circle.area());
        System.out.println("Rectangle area: " + rect.area());
    }
}
```

```
Circle area: 78.53981633974483
Rectangle area: 24.0

...Program finished with exit code 0
Press ENTER to exit console.
```

Q34.Write a program to demonstrate the use of multilevel inheritance

```java
class Shape {
    public void display() {
        System.out.println("Inside display");
    }
}
class Rectangle extends Shape {
    public void area() {
        System.out.println("Inside area");
    }
}
class Cube extends Rectangle {
    public void volume() {
        System.out.println("Inside volume");
    }
}
public class Tester {
    public static void main(String[] arguments) {
        Cube cube = new Cube();
        cube.display();
        cube.area();
        cube.volume();
    }
}
```

```
Output:

Inside display
Inside area
Inside volume
```

Q35.Write a program to demonstrate the use of multiple inheritance

```java
interface Camera
{
        void display();
    }
    interface Musicplayer
    {
        void playmusic();
    }
class Smartphone implements Camera, Musicplayer{
    public void display()
        {
            System.out.println("photo taken");
        }
        public void playmusic()
        {
            System.out.println("music played");
        }
}
class Main {
    public static void main (String[] args)
    {
        Smartphone sp=new Smartphone();
        sp.display();
        sp.playmusic();
    }
}
```

```
photo taken
music played

...Program finished with exit code 0
Press ENTER to exit console.
```

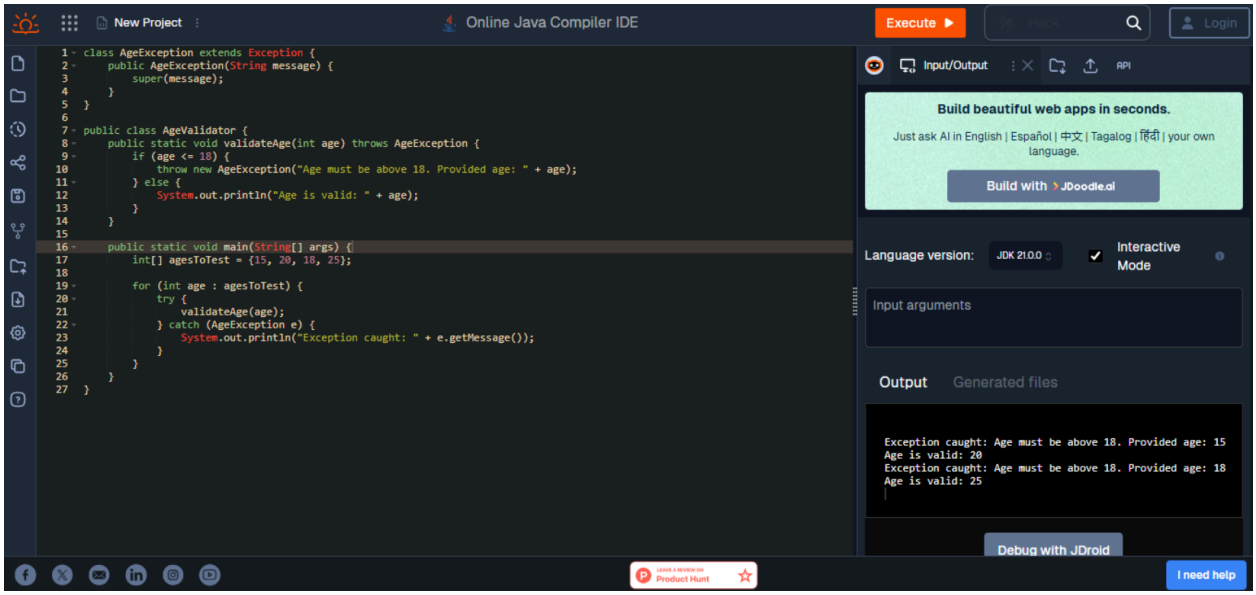
Q36.Write a Java program demonstrating String methods like substring(), replace(), and split().

```java
public class StringMethodsExample {
    public static void main(String[] args) {

        String originalString = "Hello, welcome to the world of Java programming!";

        String substringExample = originalString.substring(7, 14);
        System.out.println("Substring (7, 14): " + substringExample);

        String replaceExample = originalString.replace("Java", "Python");
        System.out.println("After replace(): " + replaceExample);

        String[] splitExample = originalString.split(" ");
        System.out.println("After split():");
        for (String word : splitExample) {
            System.out.println(word);
        }
    }
}
```

```
Substring (7, 14): welcome
After replace(): Hello, welcome to the world of Python programming!
After split():
Hello,
welcome
to
the
world
of
Java
programming!
```

Compiled and executed in 1.871 sec(s)

Q37.Create a custom exception AgeException that checks if a person's age is valid (above 18). in java



Q38.Create a Java program that demonstrates various string functions and string handling techniques in Java. This program includes common operations like:Length of a string, Concatenation, Character extraction, Substring, Searching, String comparison, Changing case, Trimming, Replacing, Splitting