MANAV RACHNA INTERNATIONAL INSTITUTE OF RESEARCH AND STUDIES

School of Computer Applications

Database Management System File
BCA-DS-155

Submitted By	
Student Name	Anushka Akharia
Roll No	23/SCA/BCA(DS)/004
Subject	Programming in Java
Semester	4th Semester
Section	BCA 4-E
Department	Computer Applications
Batch	2023-26
Faculty Name	Dr. Priyanka Sharma

Lab Session 1 – Basic Programs

1	Print "Hello World"
2	Take input and print sum of two numbers
3	Check whether a number is even or odd
4	Print sum and average of 5 numbers
5	Calculate factorial of a number
6	Print Fibonacci series up to n terms

Lab Session 2 – Number Programs

1	Reverse a number
2	Check if a number is a palindrome
3	Simple calculator
4	Check if a number is prime
5	Check if a number is an Armstrong number
6	Find the largest of three numbers using ternary operator

Lab Session 3 – Arrays & Strings (Basics)

1	Print multiplication table
2	Calculate sum and average of array elements
3	Reverse a string
4	Find factorial using recursion
5	Sort an array in ascending order

Lab Session 4 – Strings & Type Casting

1	Check palindrome for a string
2	Count vowels and consonants in a string
3	Simple banking system
4	Demonstrate type casting
5	Generate prime numbers between 1 & given number

Lab Session 5 – Classes & Methods

1	Simple class with methods
2	Class with parameterized constructor
3	Find area of a rectangle using methods
4	Bank account class with deposit and withdraw methods
5	Method overloading

Lab Session 6 – Static, Object & Inheritance

1	Demonstrate static methods
2	Method overriding
3	Getters and setters
4	Class with multiple methods
5	Object passing in methods
6	Area and perimeter using super and this keyword
7	Count number of objects using static member

Lab Session 7 – Inheritance, Interfaces & Exceptions

1	Abstract methods and abstract classes
2	Multilevel inheritance
3	Multiple inheritance
4	Partial implementation of interface
5	Custom string class: equals, reverse, case change
6	Exception handling using try and multiple catch
7	Create a package accessing external class
8	Import and use user-defined package
9	User-defined exception using throw

Lab Session 8 – GUI (AWT Basics)

1	String handling (length, concat, substring, etc.)
2	GUI component with event handling
3	(Placeholder for math calculator?)

4	Draw line, rectangle, oval, and text using graphics
5	Create a menu using frame
6	Create a dialog box
7	Implement FlowLayout and BorderLayout
8	Implement GridLayout and CardLayout
9	Frame displaying student information

Lab Session 9 – AWT Forms and Interactions

1	AWT user information form
2	AWT color picker (background color change)
3	AWT feedback form (multi-line text with buttons)

Lab Session 10 – Graphics & Layouts

1	Draw shapes (line, rectangle, oval, text)
2	Create a menu using frame
3	Create a dialog box
4	Implement FlowLayout and BorderLayout
5	Implement GridLayout and CardLayout
6	Display student information in a frame

LAB SESSION 1

1. Write a Java program to print hello world.

HelloWorld.java

```java
// Online Java Compiler
// Use this editor to write, compile and run your Java code online
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```

Output

```
Hello, World

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

2. Java program to take input from the user and print the sum of two numbers.

SumTwoNumbers.java

```java
mport java.util.Scanner;

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

        // Taking input from user
        System.out.print("Enter the first number: ");
        int num1 = scanner.nextInt();

        System.out.print("Enter the second number: ");
        int num2 = scanner.nextInt();

        // Calculating the sum
        int sum = num1 + num2;

        // Printing the result
        System.out.println("The sum is: " + sum);

        scanner.close();
    }
}
```

Output

```
ERROR!
/tmp/oe8Qbv71CS/Main.java:1: error: class, interface, enum, or record expected
mport java.util.Scanner;
^
1 error
ERROR!
error: compilation failed

=== Code Exited With Errors ===
```

3. Create a Java program to check whether a number entered by a user is even or odd.

EvenOddCheck.java

```java
import java.util.Scanner;

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

        // Taking input from user
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        // Checking even or odd
        if (number % 2 == 0) {
            System.out.println(number + " is Even.");
        } else {
            System.out.println(number + " is Odd.");
        }

        scanner.close();
    }
}
```

Output

```
Enter a number: 4
4 is Even.

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

EvenOddCheck.java

```java
import java.util.Scanner;

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

        // Taking input from user
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        // Checking even or odd
        if (number % 2 == 0) {
            System.out.println(number + " is Even.");
        } else {
            System.out.println(number + " is Odd.");
        }

        scanner.close();
    }
}
```

Output

```
Enter a number: 47
47 is Odd.

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

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

SumAndAverage.java

```java
import java.util.Scanner;

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

        int sum = 0;
        int number;
        int count = 5;

        // Taking 5 numbers as input
        for (int i = 1; i <= count; i++) {
            System.out.print("Enter number " + i + ": ");
            number = scanner.nextInt();
            sum += number;
        }

        // Calculating average
        double average = (double) sum / count;

        // Displaying the results
        System.out.println("Sum of the numbers: " + sum);
        System.out.println("Average of the numbers: " + average);

        scanner.close();
    }
}
```

Output

```
Enter number 1: 4
Enter number 2: 8
Enter number 3: 50
Enter number 4: 68
Enter number 5: 75
Sum of the numbers: 205
Average of the numbers: 41.0

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

5. Program to calculate the factorial of a number.

FactorialCalculator.java

```java
import java.util.Scanner;

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

        // Input from user
        System.out.print("Enter a non-negative integer: ");
        int number = scanner.nextInt();

        // Validate input
        if (number < 0) {
            System.out.println("Factorial is not defined for negative numbers.");
        } else {
            long factorial = 1;
            for (int i = 1; i <= number; i++) {
                factorial *= i;
            }

            // Output the result
            System.out.println("Factorial of " + number + " is: " + factorial);
        }

        scanner.close();
    }
}
```

Output

```
Enter a non-negative integer: 56
Factorial of 56 is: 6908521828386340864

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

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

FibonacciSeries.java

```java
import java.util.Scanner;

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

        // Input number of terms
        System.out.print("Enter the number of terms: ");
        int n = scanner.nextInt();

        int first = 0, second = 1;

        System.out.println("Fibonacci Series up to " + n + " terms:");

        for (int i = 1; i <= n; i++) {
            System.out.print(first + " ");

            // Generate next term
            int next = first + second;
            first = second;
            second = next;
        }

        scanner.close();
    }
}
```

Output

```
Enter the number of terms: 5
Fibonacci Series up to 5 terms:
0 1 1 2 3
=== Code Execution Successful ===
```

LAB SESSION 2

1. Program to reverse a number.

ReverseNumber.java

Share Run

```java
import java.util.Scanner;

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

        // Input number from user
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        int reversed = 0;

        // Logic to reverse the number
        while (number != 0) {
            int digit = number % 10;       // Get last digit
            reversed = reversed * 10 + digit; // Append digit
            number /= 10;                  // Remove last digit
        }

        // Output the reversed number
        System.out.println("Reversed number: " + reversed);

        scanner.close();
    }
}
```

Output

```
Enter a number: 10
Reversed number: 1

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

2. Program to check if a number is a Palindrome

PalindromeCheck.java

Share Run

```java
import java.util.Scanner;

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

        // Input from user
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        int originalNumber = number;
        int reversed = 0;

        // Reverse the number
        while (number != 0) {
            int digit = number % 10;
            reversed = reversed * 10 + digit;
            number /= 10;
        }

        // Check for palindrome
        if (originalNumber == reversed) {
            System.out.println(originalNumber + " is a palindrome.");
        } else {
            System.out.println(originalNumber + " is not a palindrome.");
        }

        scanner.close();
    }
}
```

Output

```
Enter a number: 11
11 is a palindrome.

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

PalindromeCheck.java

Share Run

```java
import java.util.Scanner;

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

        // Input from user
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        int originalNumber = number;
        int reversed = 0;

        // Reverse the number
        while (number != 0) {
            int digit = number % 10;
            reversed = reversed * 10 + digit;
            number /= 10;
        }

        // Check for palindrome
        if (originalNumber == reversed) {
            System.out.println(originalNumber + " is a palindrome.");
        } else {
            System.out.println(originalNumber + " is not a palindrome.");
        }

        scanner.close();
    }
}
```

Output

```
Enter a number: 46
46 is not a palindrome.

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

3. Program for a Simple Calculator.

SimpleCalculator.java

Share Run

```java
import java.util.Scanner;
public class SimpleCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input two numbers
        System.out.print("Enter first number: ");
        double num1 = scanner.nextDouble();

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

        // Input operation
        System.out.print("Enter an operator (+, -, *, /): ");
        char operator = scanner.next().charAt(0);

        double result;

        // Perform calculation based on operator
        switch (operator) {
            case '+':
                result = num1 + num2;
                System.out.println("Sum = " + result);
                break;
            case '-':
                result = num1 - num2;
                System.out.println("Difference = " + result);
                break;
            case '*':
                result = num1 * num2;
                System.out.println("Product = " + result);
                break;
            case '/':
                if (num2 != 0) {
                    result = num1 / num2;
                    System.out.println("Quotient = " + result);
                } else {
                    System.out.println("Error: Division by zero not allowed.");
                }
                break;
            default:
                System.out.println("Invalid operator!");
        }

        scanner.close();
    }
}
```

Output

```
Enter first number: 40
Enter second number: 109
Enter an operator (+, -, *, /): /
Quotient = 0.3669724770642202

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

4. Program to check if a Number is Prime.

PrimeCheck.java

Share Run

```java
import java.util.Scanner;

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

        // Input number from user
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        boolean isPrime = true;

        // 0 and 1 are not prime numbers
        if (number <= 1) {
            isPrime = false;
        } else {
            // Check for factors from 2 to sqrt(number)
            for (int i = 2; i <= Math.sqrt(number); i++) {
                if (number % i == 0) {
                    isPrime = false;
                    break;
                }
            }
        }

        // Output result
        if (isPrime) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }

        scanner.close();
    }
}
```

Output

```
Enter a number: 3
3 is a prime number.

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

PrimeCheck.java

Share Run

```java
import java.util.Scanner;

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

        // Input number from user
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        boolean isPrime = true;

        // 0 and 1 are not prime numbers
        if (number <= 1) {
            isPrime = false;
        } else {
            // Check for factors from 2 to sqrt(number)
            for (int i = 2; i <= Math.sqrt(number); i++) {
                if (number % i == 0) {
                    isPrime = false;
                    break;
                }
            }
        }

        // Output result
        if (isPrime) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }

        scanner.close();
    }
}
```

Output

```
Enter a number: 60
60 is not a prime number.

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

5. Program to check if a number is an Armstrong Number.

ArmstrongNumber.java

Share Run

```java
import java.util.Scanner;

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

        // Input number from user
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        int originalNumber = number;
        int sum = 0;

        // Count number of digits
        int digits = String.valueOf(number).length();

        // Calculate sum of powers of digits
        while (number != 0) {
            int digit = number % 10;
            sum += Math.pow(digit, digits);
            number /= 10;
        }

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

        scanner.close();
    }
}
```

Output

```
Enter a number: 153
153 is an Armstrong number.

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

ArmstrongNumber.java

Share Run

```java
import java.util.Scanner;

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

        // Input number from user
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        int originalNumber = number;
        int sum = 0;

        // Count number of digits
        int digits = String.valueOf(number).length();

        // Calculate sum of powers of digits
        while (number != 0) {
            int digit = number % 10;
            sum += Math.pow(digit, digits);
            number /= 10;
        }

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

        scanner.close();
    }
}
```

Output

```
Enter a number: 123
123 is not an Armstrong number.

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

6. Find the Largest of Three Numbers using Ternary Operator.

LargestUsingTernary.java

Share Run

```java
import java.util.Scanner;

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

        // Input three numbers
        System.out.print("Enter first number: ");
        int a = scanner.nextInt();

        System.out.print("Enter second number: ");
        int b = scanner.nextInt();

        System.out.print("Enter third number: ");
        int c = scanner.nextInt();

        // Find the largest using nested ternary operators
        int largest = (a > b) ? (a > c ? a : c) : (b > c ? b : c);

        // Output result
        System.out.println("The largest number is: " + largest);

        scanner.close();
    }
}
```

Output

```
Enter first number: 100
Enter second number: 0
Enter third number: 18
The largest number is: 100

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

LAB SESSION 3

1. Print Multiplication Table.

MultiplicationTable.java

```java
import java.util.Scanner;

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

        for (int i = 1; i <= 10; i++) {
            System.out.println(num + " x " + i + " = " + (num * i));
        }

        scanner.close();
    }
}
```

Output

```
Enter a number: 67
67 x 1 = 67
67 x 2 = 134
67 x 3 = 201
67 x 4 = 268
67 x 5 = 335
67 x 6 = 402
67 x 7 = 469
67 x 8 = 536
67 x 9 = 603
67 x 10 = 670

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

2. Calculate Sum and Average of Array Elements.

SumAndAverage.java

```java
import java.util.Scanner;

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

        int[] arr = new int[n];
        int sum = 0;

        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
            sum += arr[i];
        }

        double average = (double) sum / n;
        System.out.println("Sum: " + sum);
        System.out.println("Average: " + average);

        scanner.close();
    }
}
```

Output

```
Enter number of elements: 4
Enter elements:
56
80
78
40
Sum: 254
Average: 63.5

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

3. Reverse a String.

ReverseString.java

```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 str = scanner.nextLine();

        String reversed = new StringBuilder(str).reverse().toString();
        System.out.println("Reversed string: " + reversed);

        scanner.close();
    }
}
```

Output

```
Enter a string: 78
Reversed string: 87

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

4. Find Factorial of a Number using Recursion.

RecursiveFactorial.java

Share Run

```java
import java.util.Scanner;

public class RecursiveFactorial {
    public static int factorial(int n) {
        if (n == 0 || n == 1)
            return 1;
        else
            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();

        int result = factorial(number);
        System.out.println("Factorial of " + number + " is " + result);

        scanner.close();
    }
}
```

Output

```
Enter a number: 21
Factorial of 21 is -1195114496

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

5. Sort an Array in Ascending Order.

SortArray.java

Share Run

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

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

        int[] arr = new int[n];
        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        Arrays.sort(arr);

        System.out.println("Sorted array in ascending order:");
        for (int num : arr) {
            System.out.print(num + " ");
        }

        scanner.close();
    }
}
```

Output

```
Enter number of elements: 5
Enter elements:
6
80
101
-890
-67
Sorted array in ascending order:
-890 -67 6 80 101
=== Code Execution Successful ===
```

LAB SESSION 4

1. Check Palindrome for a String.

StringPalindrome.java

Share Run

```java
import java.util.Scanner;

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

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

        if (str.equals(reversed)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }

        scanner.close();
    }
}
```

Output

```
Enter a string: 56
The string is not a palindrome.

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

StringPalindrome.java

Share Run

```java
import java.util.Scanner;

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

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

        if (str.equals(reversed)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }

        scanner.close();
    }
}
```

Output

```
Enter a string: 99
The string is a palindrome.

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

2. Count Vowels and Consonants in a String.

VowelConsonantCounter.java

Share Run

```java
import java.util.Scanner;

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

        int vowels = 0, consonants = 0;

        for (char ch : str.toCharArray()) {
            if (Character.isLetter(ch)) {
                if ("aeiou".indexOf(ch) != -1)
                    vowels++;
                else
                    consonants++;
            }
        }

        System.out.println("Vowels: " + vowels);
        System.out.println("Consonants: " + consonants);

        scanner.close();
    }
}
```

Output

```
Enter a string: mississippi
Vowels: 4
Consonants: 7

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

3. Implement a Simple Banking System.

BankAccount.java

43hbp6aet

AI NEW JAVA RUN

```java
import java.util.Scanner;

// BankAccount class with deposit, withdraw, and check balance methods
class BankAccount {
    private double balance;

    // Constructor to initialize balance
    public BankAccount(double initialBalance) {
        balance = initialBalance;
    }

    // Method to deposit money
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: ₹" + amount);
        } else {
            System.out.println("Please enter a positive amount.");
        }
    }

    // Method to withdraw money
    public void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance.");
        } else if (amount <= 0) {
            System.out.println("Please enter a valid amount.");
        } else {
            balance -= amount;
            System.out.println("Withdrawn: ₹" + amount);
        }
    }

    // Method to check current balance
    public void checkBalance() {
        System.out.println("Current Balance: ₹" + balance);
    }
}

// Main class that includes the main method
public class SimpleBankingSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        BankAccount account = new BankAccount(1000);  // Starting balance

        while (true) {
            System.out.println("\n--- Banking Menu ---");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Check Balance");
            System.out.println("4. Exit");
            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double deposit = scanner.nextDouble();
                    account.deposit(deposit);
                    break;
```

STDIN

Input for the program (Optional)

Output:

```
BankAccount.java:41: error: class SimpleBankingSystem is public, should be declared in a file named SimpleBankingSystem.java
public class SimpleBankingSystem {
       ^
1 error
```

```java
                    break;

                case 2:
                    System.out.print("Enter amount to withdraw: ");
                    double withdraw = scanner.nextDouble();
                    account.withdraw(withdraw);
                    break;

                case 3:
                    account.checkBalance();
                    break;

                case 4:
                    System.out.println("Thank you for using our banking system!");
                    scanner.close();
                    return;

                default:
                    System.out.println("Invalid option. Please try again.");
            }
        }
    }
}
```

4. Write a program to demonstrate type casting.

TypeCastingDemo.java

```java
public class TypeCastingDemo {
    public static void main(String[] args) {
        // Widening (Implicit)
        int i = 10;
        double d = i;
        System.out.println("Widening Casting (int to double): " + d);

        // Narrowing (Explicit)
        double x = 9.78;
        int y = (int) x;
        System.out.println("Narrowing Casting (double to int): " + y);
    }
}
```

43hbp6aet

STDIN

Input for the program (Optional)

Output:

```
Widening Casting (int to double): 10.0
Narrowing Casting (double to int): 9
```

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

PrimeNumbersInRange.java

```java
import java.util.Scanner;

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

        System.out.println("Prime numbers between 1 and " + n + ":");
        for (int i = 2; i <= n; i++) {
            boolean isPrime = true;
            for (int j = 2; j <= Math.sqrt(i); j++) {
                if (i % j == 0) {
                    isPrime = false;
                    break;
                }
            }
            if (isPrime)
                System.out.print(i + " ");
        }

        scanner.close();
    }
}
```

43hbp6aet

STDIN

Input for the program (Optional)

Output:

```
Enter the upper limit:
Error: Command failed: timeout 7 java PrimeNumbersInRange.java
```

LAB SESSION 5

1. Program to Demonstrate a Simple Class with Methods.

SimpleClass.java

```java
class SimpleClass {
    void greet() {
        System.out.println("Hello from a simple class method!");
    }

    public static void main(String[] args) {
        SimpleClass obj = new SimpleClass();
        obj.greet();
    }
}
```

Output

```
Hello from a simple class method!

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

2. Program for Class with Parameterized Constructor.

Student.java

```java
class Student {
    String name;
    int age;

    // Parameterized constructor
    Student(String n, int a) {
        name = n;
        age = a;
    }

    void display() {
        System.out.println("Name: " + name + ", Age: " + age);
    }

    public static void main(String[] args) {
        Student s1 = new Student("Vanshika", 19);
        s1.display();
    }
}
```

Output

```
Name: Vanshika, Age: 19

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

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

Rectangle.java

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

    void setDimensions(int l, int b) {
        length = l;
        breadth = b;
    }

    int calculateArea() {
        return length * breadth;
    }

    public static void main(String[] args) {
        Rectangle rect = new Rectangle();
        rect.setDimensions(10, 5);
        System.out.println("Area of Rectangle: " + rect.calculateArea());
    }
}
```

Output

```
Area of Rectangle: 50

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

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

BankAccount.java

```java
class BankAccount {
    private double balance;

    BankAccount(double initialBalance) {
        balance = initialBalance;
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: ₹" + amount);
    }

    void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient funds.");
        } else {
            balance -= amount;
            System.out.println("Withdrawn: ₹" + amount);
        }
    }

    void showBalance() {
        System.out.println("Current Balance: ₹" + balance);
    }

    public static void main(String[] args) {
        BankAccount acc = new BankAccount(5000);
        acc.deposit(1000);
        acc.withdraw(2000);
        acc.showBalance();
    }
}
```

Output

```
Deposited: ?1000.0
Withdrawn: ?2000.0
Current Balance: ?4000.0

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

5. Program to Demonstrate Method Overloading.

Calculator.java

```java
class Calculator {
    // Overloaded add methods
    int add(int a, int b) {
        return a + b;
    }

    double add(double a, double b) {
        return a + b;
    }

    String add(String a, String b) {
        return a + b;
    }

    public static void main(String[] args) {
        Calculator calc = new Calculator();

        System.out.println("Sum of integers: " + calc.add(5, 10));
        System.out.println("Sum of doubles: " + calc.add(5.5, 4.5));
        System.out.println("Sum of strings: " + calc.add("Hello ", "World"));
    }
}
```

Output

```
Sum of integers: 15
Sum of doubles: 10.0
Sum of strings: Hello World

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

LAB SESSION 6

1. Program to Demonstrate Static Methods.

MathUtils.java

```java
class MathUtils {
    static int square(int number) {
        return number * number;
    }

    public static void main(String[] args) {
        System.out.println("Square of 5 is: " + MathUtils.square(5));
    }
}
```

Output

```
Square of 5 is: 25

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

2. Program to Demonstrate Method Overriding.

Animal.java

```java
class Animal {
    void sound() {
        System.out.println("Animal makes a sound");
    }
}

class Dog extends Animal {
    @Override
    void sound() {
        System.out.println("Dog barks");
    }

    public static void main(String[] args) {
        Animal a = new Dog();
        a.sound();
    }
}
```

Output

```
ERROR!
error: can't find main(String[]) method in class: Animal

=== Code Exited With Errors ===
```

3. Program to Demonstrate Getters and Setters.

Person.java

```java
class Person {
    private String name;

    // Setter
    public void setName(String name) {
        this.name = name;
    }

    // Getter
    public String getName() {
        return name;
    }

    public static void main(String[] args) {
        Person p = new Person();
        p.setName("Vanshika");
        System.out.println("Name is: " + p.getName());
    }
}
```

Output

```
Name is: Vanshika

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

4. Program to Demonstrate a Class with Multiple Methods.

Calculator.java

```java
class Calculator {
    int add(int a, int b) {
        return a + b;
    }

    int subtract(int a, int b) {
        return a - b;
    }

    int multiply(int a, int b) {
        return a * b;
    }

    public static void main(String[] args) {
        Calculator calc = new Calculator();
        System.out.println("Add: " + calc.add(5, 3));
        System.out.println("Subtract: " + calc.subtract(5, 3));
        System.out.println("Multiply: " + calc.multiply(5, 3));
    }
}
```

Output

```
Add: 8
Subtract: 2
Multiply: 15

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

5. Program to Demonstrate Object Passing in Methods.

Box.java

```java
class Box {
    int length;

    Box(int l) {
        length = l;
    }

    void increaseLength(Box b) {
        b.length += 10;
    }

    public static void main(String[] args) {
        Box b1 = new Box(20);
        System.out.println("Before: " + b1.length);
        b1.increaseLength(b1);
        System.out.println("After: " + b1.length);
    }
}
```

Output

```
Before: 20
After: 30

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

6. Write a program to create a Simple Class to find out the Area and Perimeter of a Rectangle using Super and this Keyword.

Shape.java

```java
class Shape {
    int length, breadth;

    Shape(int length, int breadth) {
        this.length = length;
        this.breadth = breadth;
    }
}

class Rectangle extends Shape {
    Rectangle(int length, int breadth) {
        super(length, breadth);
    }

    void calculate() {
        int area = length * breadth;
        int perimeter = 2 * (length + breadth);
        System.out.println("Area: " + area);
        System.out.println("Perimeter: " + perimeter);
    }

    public static void main(String[] args) {
        Rectangle rect = new Rectangle(10, 5);
        rect.calculate();
    }
}
```

Output

```
ERROR!
error: can't find main(String[]) method in class: Shape

=== Code Exited With Errors ===
```

7. Write a Program to Count the Number of Objects Created for a Class using Static Member Function.

Counter.java

Share Run

```java
class Counter {
    static int count = 0;

    Counter() {
        count++;
    }

    static void displayCount() {
        System.out.println("Objects created: " + count);
    }

    public static void main(String[] args) {
        Counter c1 = new Counter();
        Counter c2 = new Counter();
        Counter c3 = new Counter();

        Counter.displayCount();
    }
}
```

Output

```
Objects created: 3

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

LAB SESSION 7

1. Write a Program to Design a Class using Abstract Methods and Abstract Classes.

Circle.java

```java
abstract class Shape {
    abstract void draw();
}

class Circle extends Shape {
    void draw() {
        System.out.println("Drawing a circle.");
    }

    public static void main(String[] args) {
        Shape obj = new Circle();
        obj.draw();
    }
}
```

Output

```
ERROR!
error: can't find main(String[]) method in class: Shape

=== Code Exited With Errors ===
```

2. Write a Program to Demonstrate the use of Multilevel Inheritance.

Animal.java

```java
class Animal {
    void eat() {
        System.out.println("This animal eats food.");
    }
}

class Dog extends Animal {
    void bark() {
        System.out.println("Dog barks.");
    }
}

class Puppy extends Dog {
    void weep() {
        System.out.println("Puppy weeps.");
    }

    public static void main(String[] args) {
        Puppy p = new Puppy();
        p.eat();
        p.bark();
        p.weep();
    }
}
```

Output

```
ERROR!
error: can't find main(String[]) method in class: Animal

=== Code Exited With Errors ===
```

3. Write a Program that shows Partial Implementation of Interface.

CompleteImpl.java

```java
interface MyInterface {
    void method1();
    void method2();
}

abstract class PartialImpl implements MyInterface {
    public void method1() {
        System.out.println("Method1 implemented.");
    }
}

class CompleteImpl extends PartialImpl {
    public void method2() {
        System.out.println("Method2 implemented.");
    }

    public static void main(String[] args) {
        CompleteImpl obj = new CompleteImpl();
        obj.method1();
        obj.method2();
    }
}
```

Output

```
ERROR!
error: can't find main(String[]) method in class: MyInterface

=== Code Exited With Errors ===
```

4. Write a Program to Design a String Class that performs the String Method [Equal, Reverse the String, Change Case].

MyString.java

```java
class MyString {
    String str;

    MyString(String s) {
        str = s;
    }

    void checkEqual(String other) {
        System.out.println("Strings are " + (str.equals(other) ? "Equal" : "Not Equal"));
    }

    void reverse() {
        StringBuilder sb = new StringBuilder(str);
        System.out.println("Reversed: " + sb.reverse());
    }

    void changeCase() {
        System.out.println("Uppercase: " + str.toUpperCase());
        System.out.println("Lowercase: " + str.toLowerCase());
    }

    public static void main(String[] args) {
        MyString ms = new MyString("HelloWorld");
        ms.checkEqual("HelloWorld");
        ms.reverse();
        ms.changeCase();
    }
}
```

Output

```
Strings are Equal
Reversed: dlroWolleH
Uppercase: HELLOWORLD
Lowercase: helloworld

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

5. Write a Program to handle the Exception using Try and Multiple Catch Block.

MultiCatchExample.java

```java
public class MultiCatchExample {
    public static void main(String[] args) {
        try {
            int[] arr = {1, 2, 3};
            System.out.println(arr[5]);
            int result = 10 / 0;
        } catch (ArithmeticException e) {
            System.out.println("Arithmetic error: " + e);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Array index error: " + e);
        } catch (Exception e) {
            System.out.println("General error: " + e);
        }
    }
}
```

Output

```
ERROR!
Array index error: java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 3

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

6. Write a program to create a package that access the member of External class as well as same package.

ExternalClass.java

```
package myPackage;

public class ExternalClass {
    public void show() {
        System.out.println("Accessing External Class from same package.");
    }
}
```

Output

```
ERROR!
error: can't find main(String[]) method in class: myPackage.ExternalClass

=== Code Exited With Errors ===
```

MainClass.java

```
package myPackage;

public class MainClass {
    public static void main(String[] args) {
        ExternalClass obj = new ExternalClass();
        obj.show();
    }
}
```

Output

```
ERROR!
/tmp/uCHvGYR1wz/Main.java:5: error: cannot find symbol
        ExternalClass obj = new ExternalClass();
        ^
  symbol:   class ExternalClass
  location: class MainClass
ERROR!
/tmp/uCHvGYR1wz/Main.java:5: error: cannot find symbol
        ExternalClass obj = new ExternalClass();
                            ^
  symbol:   class ExternalClass
  location: class MainClass
2 errors
ERROR!
error: compilation failed

=== Code Exited With Errors ===
```

7. Write a program that import the user define package and access the Member variable of classes that contained by package.

MyUtil.java

```
package utilities;

public class MyUtil {
    public int number = 42;
}
```

Output

```
ERROR!
error: can't find main(String[]) method in class: utilities.MyUtil

=== Code Exited With Errors ===
```

MainApp.java

```
package app;
import utilities.MyUtil;

public class MainApp {
    public static void main(String[] args) {
        MyUtil util = new MyUtil();
        System.out.println("Accessed number: " + util.number);
    }
}
```

Output

```
ERROR!
/tmp/ieVIaiX1Q5/Main.java:2: error: package utilities does not exist
import utilities.MyUtil;
               ^
ERROR!
/tmp/ieVIaiX1Q5/Main.java:6: error: cannot find symbol
        MyUtil util = new MyUtil();
        ^
  symbol:   class MyUtil
  location: class MainApp
ERROR!
/tmp/ieVIaiX1Q5/Main.java:6: error: cannot find symbol
        MyUtil util = new MyUtil();
                      ^
  symbol:   class MyUtil
  location: class MainApp
3 errors
ERROR!
error: compilation failed

=== Code Exited With Errors ===
```

8. Write a program to handle the user defined exception using throw keyword.

InvalidAgeException.java

Share Run

```java
class InvalidAgeException extends Exception {
    InvalidAgeException(String msg) {
        super(msg);
    }
}

public class CustomExceptionDemo {
    static void validateAge(int age) throws InvalidAgeException {
        if (age < 18)
            throw new InvalidAgeException("Age must be 18 or above.");
        else
            System.out.println("Valid age: " + age);
    }

    public static void main(String[] args) {
        try {
            validateAge(16);
        } catch (InvalidAgeException e) {
            System.out.println("Caught Exception: " + e.getMessage());
        }
    }
}
```

Output

```
ERROR!
error: can't find main(String[]) method in class: InvalidAgeException

=== Code Exited With Errors ===
```

LAB SESSION 8

1. 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.

StringFunctions.java

```java
public class StringFunctions {
    public static void main(String[] args) {
        String str1 = "  Hello ";
        String str2 = "World";

        System.out.println("Length: " + str1.length());
        System.out.println("Concatenation: " + str1.concat(str2));
        System.out.println("Character at 1st position: " + str1.charAt(0));
        System.out.println("Substring (1 to 4): " + str1.substring(1, 4));
        System.out.println("Contains 'lo': " + str1.contains("lo"));
        System.out.println("Compare to 'hello': " + str1.trim().compareTo("hello"));
        System.out.println("To Upper Case: " + str1.toUpperCase());
        System.out.println("To Lower Case: " + str1.toLowerCase());
        System.out.println("Trimmed: '" + str1.trim() + "'");
        System.out.println("Replace 'l' with 'p': " + str1.replace('l', 'p'));

        String str3 = "This is Java";
        String[] words = str3.split(" ");
        System.out.println("Split:");
        for (String word : words)
            System.out.println(word);
    }
}
```

Output

```
Length: 8
Concatenation:   Hello World
Character at 1st position:
Substring (1 to 4):  He
Contains 'lo': true
Compare to 'hello': -32
To Upper Case:   HELLO
To Lower Case:   hello
Trimmed: 'Hello'
Replace 'l' with 'p':   Heppo
Split:
This
is
Java

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

2. Write a Program to create a Class Component that Shows Controls and Event Handling on that Controls.

Main.java

```java
import java.awt.*;
import java.awt.event.*;

public class ControlDemo extends Frame implements ActionListener {
    TextField tf;
    Button b;

    ControlDemo() {
        tf = new TextField();
        tf.setBounds(60, 50, 170, 20);

        b = new Button("Click");
        b.setBounds(100, 120, 80, 30);
        b.addActionListener(this);

        add(tf);
        add(b);

        setSize(300, 300);
        setLayout(null);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        tf.setText("Welcome to Java!");
    }

    public static void main(String[] args) {
        new ControlDemo();
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at java.desktop/java.awt.Frame.<init>(Frame.java:393)
    at ControlDemo.<init>(Main.java:8)
    at ControlDemo.main(Main.java:29)

=== Code Exited With Errors ===
```

3. Write a Program to Draw the Line, Rectangle, Oval, Text using the Graphics Method.

DrawingExample.java

```java
import java.awt.*;

public class DrawingExample extends Frame {
    public void paint(Graphics g) {
        g.drawLine(30, 50, 100, 50);
        g.drawRect(30, 70, 100, 50);
        g.drawOval(30, 130, 100, 50);
        g.drawString("Graphics in Java", 30, 200);
    }

    public static void main(String[] args) {
        DrawingExample d = new DrawingExample();
        d.setSize(300, 300);
        d.setTitle("Draw Shapes");
        d.setVisible(true);
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at java.desktop/java.awt.Frame.<init>(Frame.java:393)
    at DrawingExample.<init>(Main.java:3)
    at DrawingExample.main(Main.java:12)

=== Code Exited With Errors ===
```

4. Write a Program to Create a Menu using the Frame.

MenuExample.java

```java
import java.awt.*;
import java.awt.event.*;

public class MenuExample extends Frame {
    MenuExample() {
        MenuBar mb = new MenuBar();
        Menu menu = new Menu("Menu");
        MenuItem i1 = new MenuItem("Item 1");
        MenuItem i2 = new MenuItem("Item 2");
        menu.add(i1);
        menu.add(i2);
        mb.add(menu);
        setMenuBar(mb);

        setSize(300, 300);
        setLayout(null);
        setVisible(true);
    }

    public static void main(String[] args) {
        new MenuExample();
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at java.desktop/java.awt.Frame.<init>(Frame.java:393)
    at MenuExample.<init>(Main.java:5)
    at MenuExample.main(Main.java:21)

=== Code Exited With Errors ===
```

5. Write a Program to Create a Dialog Box.

DialogExample.java

```java
import java.awt.*;
import java.awt.event.*;

public class DialogExample {
    DialogExample() {
        Frame f = new Frame();
        Dialog d = new Dialog(f, "Dialog Example", true);
        d.setLayout(new FlowLayout());
        Button b = new Button("Close");
        b.addActionListener(e -> d.setVisible(false));
        d.add(new Label("This is a dialog"));
        d.add(b);
        d.setSize(200, 100);
        d.setVisible(true);
    }

    public static void main(String[] args) {
        new DialogExample();
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at java.desktop/java.awt.Frame.<init>(Frame.java:393)
    at DialogExample.<init>(Main.java:6)
    at DialogExample.main(Main.java:18)

=== Code Exited With Errors ===
```

6. Write a Program to Implement the FlowLayout and BorderLayout.

LayoutExample.java

```java
import java.awt.*;

public class LayoutExample {
    public static void main(String[] args) {
        Frame f = new Frame("Flow & Border Layout");

        // FlowLayout
        f.setLayout(new FlowLayout());
        f.add(new Button("One"));
        f.add(new Button("Two"));
        f.add(new Button("Three"));
        f.setSize(300, 100);
        f.setVisible(true);

        // For BorderLayout, comment FlowLayout and use below:
        /*
        f.setLayout(new BorderLayout());
        f.add(new Button("North"), BorderLayout.NORTH);
        f.add(new Button("South"), BorderLayout.SOUTH);
        f.add(new Button("East"), BorderLayout.EAST);
        f.add(new Button("West"), BorderLayout.WEST);
        f.add(new Button("Center"), BorderLayout.CENTER);
        f.setSize(300, 200);
        f.setVisible(true);
        */
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at LayoutExample.main(Main.java:5)

=== Code Exited With Errors ===
```

7. Write a Program to Implement the GridLayout, CardLayout.

GridCardExample.java

```java
import java.awt.*;

public class GridCardExample {
    public static void main(String[] args) {
        Frame f = new Frame("Layouts");

        // GridLayout
        f.setLayout(new GridLayout(2, 2));
        f.add(new Button("1"));
        f.add(new Button("2"));
        f.add(new Button("3"));
        f.add(new Button("4"));
        f.setSize(200, 200);
        f.setVisible(true);

        // CardLayout can be done in another Frame
        /*
        Frame f2 = new Frame("CardLayout Example");
        CardLayout cl = new CardLayout();
        f2.setLayout(cl);

        Button b1 = new Button("Card1");
        Button b2 = new Button("Card2");

        f2.add("First", b1);
        f2.add("Second", b2);

        cl.show(f2, "First");
        f2.setSize(200, 200);
        f2.setVisible(true);
        */
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at GridCardExample.main(Main.java:5)

=== Code Exited With Errors ===
```

8. Write a Program to Create Frame that Display the Student Information

StudentInfoFrame.java

```java
import java.awt.*;

public class StudentInfoFrame {
    StudentInfoFrame() {
        Frame f = new Frame("Student Information");

        Label name = new Label("Name: Vanshika Goyal");
        Label roll = new Label("Roll No: 101");
        Label course = new Label("Course: B.Tech CSE");

        name.setBounds(50, 50, 200, 20);
        roll.setBounds(50, 80, 200, 20);
        course.setBounds(50, 110, 200, 20);

        f.add(name);
        f.add(roll);
        f.add(course);

        f.setSize(300, 200);
        f.setLayout(null);
        f.setVisible(true);
    }

    public static void main(String[] args) {
        new StudentInfoFrame();
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at StudentInfoFrame.<init>(Main.java:5)
    at StudentInfoFrame.main(Main.java:25)

=== Code Exited With Errors ===
```

.

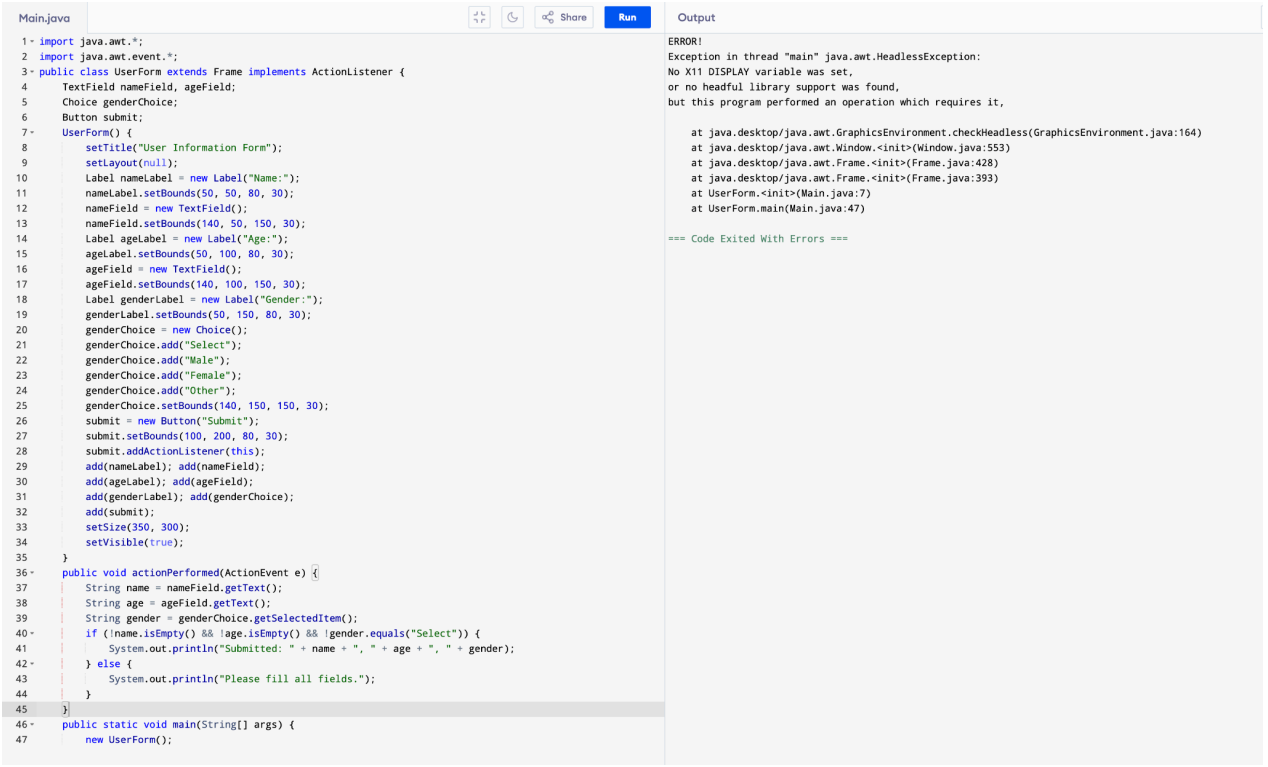
LAB SESSION 9

1. AWT User Information Form

- 🔹 Description: Collects user Data like Name, Age, and Gender.

🧩 Features: Input fields and Drop-Down for Gender.

Shows Message on Submission.



Main.java

Share | Run

```java
import java.awt.*;
import java.awt.event.*;
public class UserForm extends Frame implements ActionListener {
    TextField nameField, ageField;
    Choice genderChoice;
    Button submit;
    UserForm() {
        setTitle("User Information Form");
        setLayout(null);
        Label nameLabel = new Label("Name:");
        nameLabel.setBounds(50, 50, 80, 30);
        nameField = new TextField();
        nameField.setBounds(140, 50, 150, 30);
        Label ageLabel = new Label("Age:");
        ageLabel.setBounds(50, 100, 80, 30);
        ageField = new TextField();
        ageField.setBounds(140, 100, 150, 30);
        Label genderLabel = new Label("Gender:");
        genderLabel.setBounds(50, 150, 80, 30);
        genderChoice = new Choice();
        genderChoice.add("Select");
        genderChoice.add("Male");
        genderChoice.add("Female");
        genderChoice.add("Other");
        genderChoice.setBounds(140, 150, 150, 30);
        submit = new Button("Submit");
        submit.setBounds(100, 200, 80, 30);
        submit.addActionListener(this);
        add(nameLabel); add(nameField);
        add(ageLabel); add(ageField);
        add(genderLabel); add(genderChoice);
        add(submit);
        setSize(350, 300);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        String name = nameField.getText();
        String age = ageField.getText();
        String gender = genderChoice.getSelectedItem();
        if (!name.isEmpty() && !age.isEmpty() && !gender.equals("Select")) {
            System.out.println("Submitted: " + name + ", " + age + ", " + gender);
        } else {
            System.out.println("Please fill all fields.");
        }
    }
    public static void main(String[] args) {
        new UserForm();
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at java.desktop/java.awt.Frame.<init>(Frame.java:393)
    at UserForm.<init>(Main.java:7)
    at UserForm.main(Main.java:47)

=== Code Exited With Errors ===
```

2. AWT Color Picker

- Description: Lets the User Pick a Background Color for the Window.

🧩 Features: Radio Buttons or Drop-Down for Color choices.

Changes Background based on Selection.

Uses CheckboxGroup or Choice, Button.

Main.java

Share Run

```java
import java.awt.*;
import java.awt.event.*;
public class ColorPicker extends Frame implements ItemListener {
    Choice colorChoice;
    Button apply;
    ColorPicker() {
        setTitle("AWT Color Picker");
        setLayout(null);
        Label label = new Label("Pick a Color:");
        label.setBounds(50, 50, 100, 30);
        colorChoice = new Choice();
        colorChoice.add("White");
        colorChoice.add("Red");
        colorChoice.add("Green");
        colorChoice.add("Blue");
        colorChoice.setBounds(160, 50, 100, 30);
        colorChoice.addItemListener(this);
        add(label);
        add(colorChoice);
        setSize(300, 200);
        setVisible(true);
    }
    public void itemStateChanged(ItemEvent e) {
        String color = colorChoice.getSelectedItem();
        switch (color) {
            case "Red":
                setBackground(Color.RED);
                break;
            case "Green":
                setBackground(Color.GREEN);
                break;
            case "Blue":
                setBackground(Color.BLUE);
                break;
            default:
                setBackground(Color.WHITE);
        }
    }
    public static void main(String[] args) {
        new ColorPicker();
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at java.desktop/java.awt.Frame.<init>(Frame.java:393)
    at ColorPicker.<init>(Main.java:8)
    at ColorPicker.main(Main.java:48)

=== Code Exited With Errors ===
```

3. AWT Feedback Form

- Description: A form to Collect Feedback or Suggestions from Users.

🧩 Features: Multi-line Text Input using TextArea.

Submit and Clear Buttons.

Displays Submitted Feedback or Stores in Memory.

Main.java

Share | Run | Output

```java
import java.awt.*;
import java.awt.event.*;

public class FeedbackForm extends Frame implements ActionListener {
    TextArea feedbackArea;
    Button submit, clear;
    Label displayLabel;

    FeedbackForm() {
        setTitle("AWT Feedback Form");
        setLayout(null);
        Label label = new Label("Your Feedback:");
        label.setBounds(50, 50, 100, 30);
        feedbackArea = new TextArea();
        feedbackArea.setBounds(50, 90, 250, 100);
        submit = new Button("Submit");
        submit.setBounds(50, 200, 80, 30);
        clear = new Button("Clear");
        clear.setBounds(140, 200, 80, 30);
        submit.addActionListener(this);
        clear.addActionListener(this);
        displayLabel = new Label();
        displayLabel.setBounds(50, 240, 300, 30);
        add(label);
        add(feedbackArea);
        add(submit);
        add(clear);
        add(displayLabel);
        setSize(350, 300);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == submit) {
            String feedback = feedbackArea.getText();
            displayLabel.setText("Submitted: " + feedback);
        } else if (e.getSource() == clear) {
            feedbackArea.setText("");
            displayLabel.setText("");
        }
    }
    public static void main(String[] args) {
        new FeedbackForm();
    }
}
```

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at java.desktop/java.awt.Frame.<init>(Frame.java:393)
    at FeedbackForm.<init>(Main.java:9)
    at FeedbackForm.main(Main.java:50)

=== Code Exited With Errors ===
```

LAB SESSION 10

1. Write a program to Draw the Line, Rectangle, Oval, Text using the Graphics Method.

DrawingExample.java

```java
import java.awt.*;

public class DrawingExample extends Frame {
    public void paint(Graphics g) {
        g.drawLine(30, 50, 100, 50);
        g.drawRect(30, 70, 100, 50);
        g.drawOval(30, 130, 100, 50);
        g.drawString("Graphics in Java", 30, 200);
    }

    public static void main(String[] args) {
        DrawingExample d = new DrawingExample();
        d.setSize(300, 300);
        d.setTitle("Draw Shapes");
        d.setVisible(true);
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at java.desktop/java.awt.Frame.<init>(Frame.java:393)
    at DrawingExample.<init>(Main.java:3)
    at DrawingExample.main(Main.java:12)

=== Code Exited With Errors ===
```

2. Write a Program to Create a Menu Using the Frame.

MenuExample.java

```java
import java.awt.*;
import java.awt.event.*;

public class MenuExample extends Frame {
    MenuExample() {
        MenuBar mb = new MenuBar();
        Menu menu = new Menu("Menu");
        MenuItem i1 = new MenuItem("Item 1");
        MenuItem i2 = new MenuItem("Item 2");
        menu.add(i1);
        menu.add(i2);
        mb.add(menu);
        setMenuBar(mb);

        setSize(300, 300);
        setLayout(null);
        setVisible(true);
    }

    public static void main(String[] args) {
        new MenuExample();
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at java.desktop/java.awt.Frame.<init>(Frame.java:393)
    at MenuExample.<init>(Main.java:5)
    at MenuExample.main(Main.java:21)

=== Code Exited With Errors ===
```

3. Write a Program to Create a Dialog Box.

DialogExample.java

```java
import java.awt.*;
import java.awt.event.*;

public class DialogExample {
    DialogExample() {
        Frame f = new Frame();
        Dialog d = new Dialog(f, "Dialog Example", true);
        d.setLayout(new FlowLayout());
        Button b = new Button("Close");
        b.addActionListener(e -> d.setVisible(false));
        d.add(new Label("This is a dialog"));
        d.add(b);
        d.setSize(200, 100);
        d.setVisible(true);
    }

    public static void main(String[] args) {
        new DialogExample();
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at java.desktop/java.awt.Frame.<init>(Frame.java:393)
    at DialogExample.<init>(Main.java:6)
    at DialogExample.main(Main.java:18)

=== Code Exited With Errors ===
```

4. Write a Program to Implement the FlowLayout and BorderLayout.

LayoutExample.java

```java
import java.awt.*;

public class LayoutExample {
    public static void main(String[] args) {
        Frame f = new Frame("Flow & Border Layout");

        // FlowLayout
        f.setLayout(new FlowLayout());
        f.add(new Button("One"));
        f.add(new Button("Two"));
        f.add(new Button("Three"));
        f.setSize(300, 100);
        f.setVisible(true);

        // For BorderLayout, comment FlowLayout and use below:
        /*
        f.setLayout(new BorderLayout());
        f.add(new Button("North"), BorderLayout.NORTH);
        f.add(new Button("South"), BorderLayout.SOUTH);
        f.add(new Button("East"), BorderLayout.EAST);
        f.add(new Button("West"), BorderLayout.WEST);
        f.add(new Button("Center"), BorderLayout.CENTER);
        f.setSize(300, 200);
        f.setVisible(true);
        */
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at LayoutExample.main(Main.java:5)

=== Code Exited With Errors ===
```

5. Write a Program to Implement the GridLayout, CardLayout.

GridCardExample.java

```java
import java.awt.*;

public class GridCardExample {
    public static void main(String[] args) {
        Frame f = new Frame("Layouts");

        // GridLayout
        f.setLayout(new GridLayout(2, 2));
        f.add(new Button("1"));
        f.add(new Button("2"));
        f.add(new Button("3"));
        f.add(new Button("4"));
        f.setSize(200, 200);
        f.setVisible(true);

        // CardLayout can be done in another Frame
        /*
        Frame f2 = new Frame("CardLayout Example");
        CardLayout cl = new CardLayout();
        f2.setLayout(cl);

        Button b1 = new Button("Card1");
        Button b2 = new Button("Card2");

        f2.add("First", b1);
        f2.add("Second", b2);

        cl.show(f2, "First");
        f2.setSize(200, 200);
        f2.setVisible(true);
        */
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at GridCardExample.main(Main.java:5)

=== Code Exited With Errors ===
```

6. Write a Program to Create a Frame that displays the Student Information.

StudentInfoFrame.java

```java
import java.awt.*;

public class StudentInfoFrame {
    StudentInfoFrame() {
        Frame f = new Frame("Student Information");

        Label name = new Label("Name: Vanshika Goyal");
        Label roll = new Label("Roll No: 101");
        Label course = new Label("Course: B.Tech CSE");

        name.setBounds(50, 50, 200, 20);
        roll.setBounds(50, 80, 200, 20);
        course.setBounds(50, 110, 200, 20);

        f.add(name);
        f.add(roll);
        f.add(course);

        f.setSize(300, 200);
        f.setLayout(null);
        f.setVisible(true);
    }

    public static void main(String[] args) {
        new StudentInfoFrame();
    }
}
```

Output

```
ERROR!
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set,
or no headful library support was found,
but this program performed an operation which requires it,

    at java.desktop/java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:164)
    at java.desktop/java.awt.Window.<init>(Window.java:553)
    at java.desktop/java.awt.Frame.<init>(Frame.java:428)
    at StudentInfoFrame.<init>(Main.java:5)
    at StudentInfoFrame.main(Main.java:25)

=== Code Exited With Errors ===
```