SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

SRM INSTITUTE OF SCIENCE & TECHNOLOGY, NCR CAMPUS, MODINAGAR

(FACULTY OF SCIENCE AND HUMANITIES)

DEPARTMENT OF COMPUTER APPLICATIONS

PRACTICAL FILE

Programming Using Java [PCA20C01J]

MCA 1ST YEAR, 1ST SEMESTER

Session: July- Nov, 2023

Submitted To:	**Submitted By:**
Ms. Vaishali Gupta	**ANIKET CHANDELA**
(Assistant Professor)	**RA2332241030096**

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

SRM INSTITUTE OF SCIENCE & TECHNOLOGY, NCR CAMPUS, MODINAGAR

(FACULTY OF SCIENCE AND HUMANITIES) DEPARTMENT OF COMPUTER APPLICATIONS

Registration No.: RA2332241030096

BONAFIDE CERTIFICATE

Certified to be the bonafide record of the work done by **ANIKET CHANDELA** of MCA, First year,First Semester(section B) for the award of **Masters** degree course in the FACULTY OF SCIENCE & HUMANITIES in DEPARTMENT OF COMPUTER APPLICATIONS
in **Programming Using Java [PCA20C01J]** laboratory during the Academic year-2023-24.

Subject In-Charge **HEAD OF THE DEPARTMENT**

Submitted for the university examination held on ________________

INTERNAL EXAMINER 1 **INTERNAL EXAMINER 2**

INDEX

9	Create a class “Enclosed” within it create inner class “Nested”, both the classes should have at least one method to display messages. Try to call the method of the “Nested” class in the “Enclosed” class and vice versa.	24	23/08/2023	
10	Create a class called MyString : Declare two string type variables: str1 (“ Welcome to Java tutorial”) and str2(“Today's topic is String Handling in Java”). Perform following operations in this class: i. Concatenate two strings ii. Covert str1 into lower case iii. Covert str2 into upper case iv. Are both equal to each other v. Show the location of “J” in both str1 and str2 vi. Replace “i” with “I” in both the strings vii. display “java” from str string viii. Display the “7” character in str1. ix. Convert str1 into string array	26	23/08/2023	
11	Create a class person (Data Member: Name & address , Method: Accept() and display() to accept and display value of data member on Output device. Derive two classes student (Data Member: Rollno, Course Member Method: Accept() and Display()) and Employee((Data Member: EmpId ,Department Member Method: Accept() and Display()).display details of one student and one employee. NOTE: use super keyword to invoke hidden members of base class.	28	**30/08/2023**	
12	WAP to show the use of Interfaces in java.	33	**30/08/2023**	
13	Write a java program to display the grade of students depending on marks, please raise a user defined checked Exception, if less than 0 or more than 100 marks are entered for grade.	36	06/09/2023	
14	Write a Java program to use the try and catch and finally block.	38	06/09/2023	

15	Write a multithreaded program where one thread will print 0-5 and another thread will print 5-0. Use thread class.	40	06/09/2023	
16	Write a java multithreaded (2 or more)java program , one thread will print odd numbers and another will print even numbers and the main thread is there it will print date and time. Use Runnable interface	42	13/09/2023	
17	WAP to show the use of Legacy classes:- Vector	44	13/09/2023	
18	WAP to show the use of Legacy classes:- Stack	46	20/09/2023	
19	WAP to demonstrate the use of followings: i. StringTokenizer ii. Date iii. Calendar	47	20/09/2023	
20	WAP to create a Simple GUI with text field button and label and handle click event of button.	49	27/09/2023	
21	WAP to show different layouts using AWT controls.	51	27/09/2023	
22	WAP to create a GUI to show checkboxes handling their events.	54	11/09/2023	
23	WAP to show the use of Console class for reading and writing.	57	11/09/2023	
24	WAP to count the number of characters , words and lines in a file.	59	18/10/2023	

Program 1

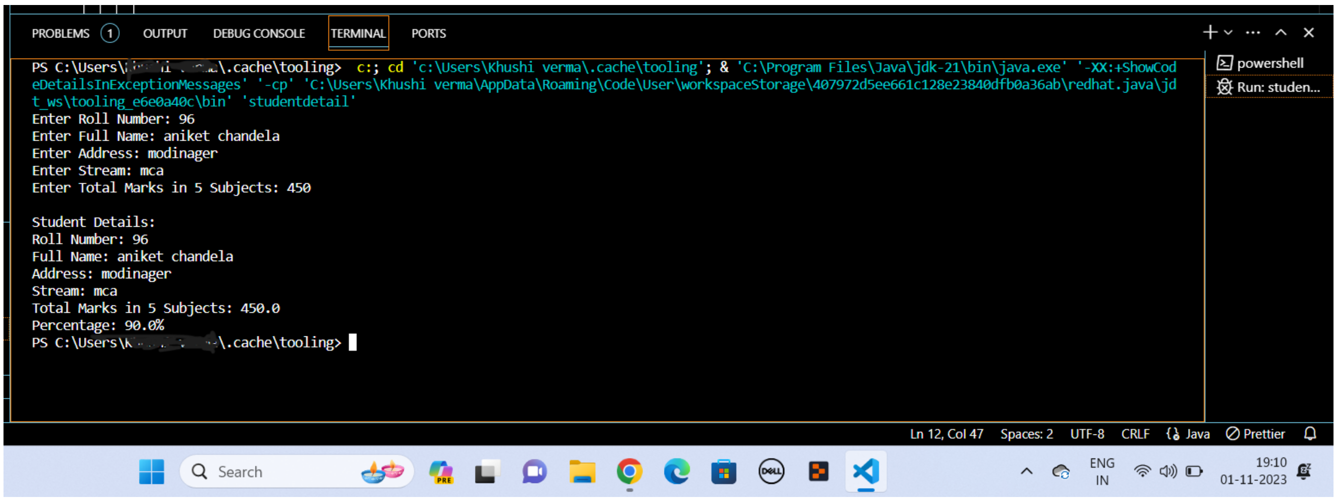
Aim: Write a Java program to accept following details about a student as follows:

i. rollno
ii. Fullname
iii. Address
iv. Stream
v. total marks in 5 subjects
vi. percentage display all the details in a readable format?

Code:

```
import java.util.Scanner;


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


        // Accept student details
        System.out.print("Enter Roll Number: ");
int rollNo = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character


        System.out.print("Enter Full Name: ");
        String fullName = scanner.nextLine();


        System.out.print("Enter Address: ");
        String address = scanner.nextLine();


        System.out.print("Enter Stream: ");
        String stream = scanner.nextLine();
        System.out.print("Enter Total Marks in 5 Subjects: ");
double totalMarks = scanner.nextDouble();

        // Calculate percentage
        double percentage = (totalMarks / 500) * 100;

        // Display the student details
        System.out.println("\nStudent Details:");
        System.out.println("Roll Number: " + rollNo);
        System.out.println("Full Name: " + fullName);
        System.out.println("Address: " + address);
        System.out.println("Stream: " + stream);
        System.out.println("Total Marks in 5 Subjects: " + totalMarks);
        System.out.println("Percentage: " + percentage + "%");

        scanner.close();
    } }
```

Output:



```
PS C:\Users\[redacted]\.cache\tooling>  c:; cd 'c:\Users\Khushi verma\.cache\tooling'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCod
eDetailsInExceptionMessages' '-cp' 'C:\Users\Khushi verma\AppData\Roaming\Code\User\workspaceStorage\407972d5ee661c128e23840dfb0a36ab\redhat.java\jd
t_ws\tooling_e6e0a40c\bin' 'studentdetail'
Enter Roll Number: 96
Enter Full Name: aniket chandela
Enter Address: modinager
Enter Stream: mca
Enter Total Marks in 5 Subjects: 450

Student Details:
Roll Number: 96
Full Name: aniket chandela
Address: modinager
Stream: mca
Total Marks in 5 Subjects: 450.0
Percentage: 90.0%
PS C:\Users\[redacted]\.cache\tooling>
```

Program 2

Aim: Write a java program to print following output:

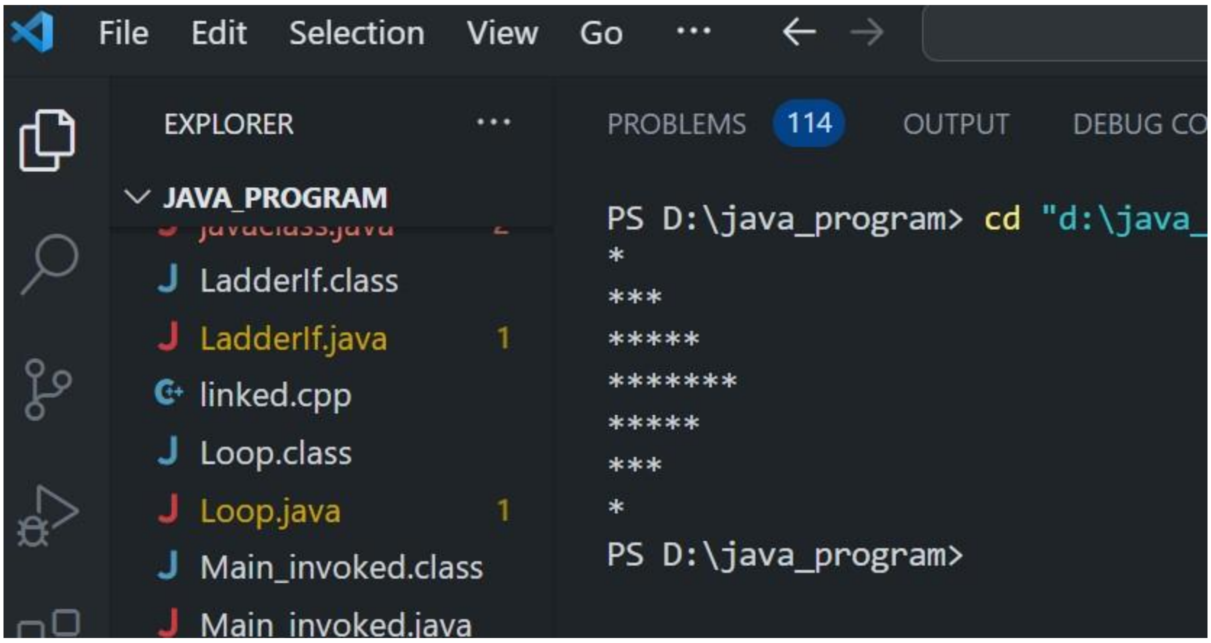
```
*
***
*****
*******
*****
***
*
```

Code:

```
public class Main {
  public static void main(String[] args) {
      int n = 4; // Number of rows in the upper half of the pattern

      // Print upper half of the pattern
for (int i = 1; i <= n; i++) {
 for (int j = 1; j <= 2 * i - 1; j++) {
          System.out.print("*");
        }
        System.out.println();
      }

      // Print lower half of the pattern
for (int i = n - 1; i >= 1; i--) {
  for (int j = 1; j <= 2 * i - 1; j++) {
          System.out.print("*");
        }
        System.out.println();
      }
```

```
    }
}
```

Output:



```
PS D:\java_program> cd "d:\java_
*
***
*****
*******
*****
***
*
PS D:\java_program>
```

Program 3

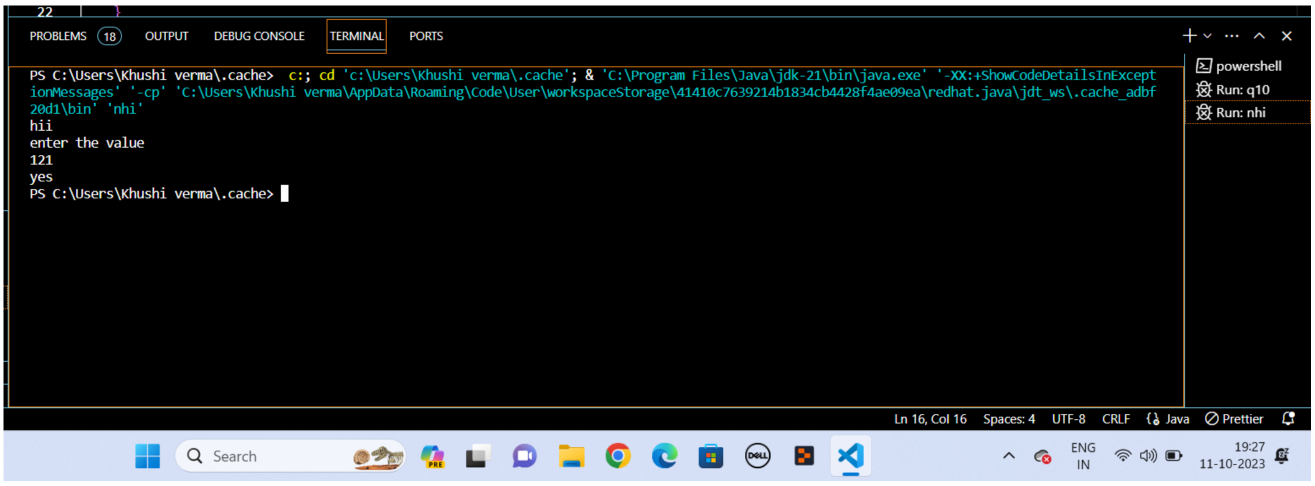
Aim: Write a java program to check if a number entered by the user is “palindrome” or not.

Code:

```
import java.util.Scanner;

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

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

        // Check if it's a palindrome
        boolean isPalindrome = isPalindrome(number);

        // Display the result
if (isPalindrome) {
            System.out.println(number + " is a palindrome.");
        } else {
            System.out.println(number + " is not a palindrome.");
        }

        scanner.close();
    }

    // Function to check if a number is a palindrome
public static boolean isPalindrome(int num) {
    int originalNumber = num;
```

```
 int reversedNumber = 0;

        while (num > 0) {
int digit = num % 10;
            reversedNumber = reversedNumber * 10 + digit;
num /= 10;
        }

        return originalNumber == reversedNumber;
    }
}
```

Output:



```
PS C:\Users\Khushi verma\.cache>  c:; cd 'c:\Users\Khushi verma\.cache'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Khushi verma\AppData\Roaming\Code\User\workspaceStorage\41410c7639214b1834cb4428f4ae09ea\redhat.java\jdt_ws\.cache_adbf20d1\bin' 'nhi'
hii
enter the value
121
yes
PS C:\Users\Khushi verma\.cache> 
```

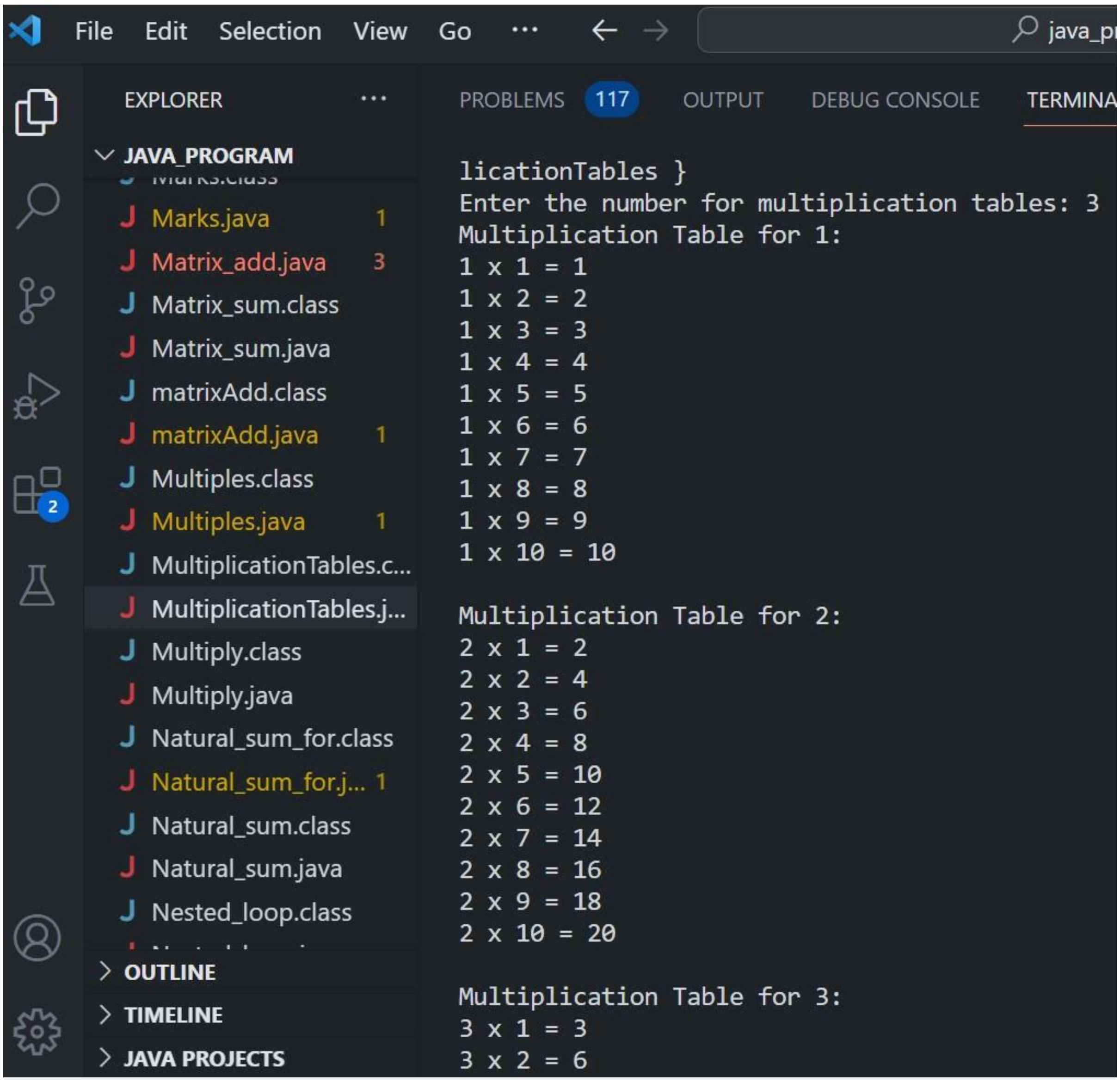
Program 4

Aim: Write a java program to print tables from 0 to accepted numbers, using loops and keyboard inputs.

Code:

```
import java.util.Scanner;


public class MultiplicationTables {
public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number for multiplication tables: ");
int num = scanner.nextInt();         for (int i = 0; i <= num; i++) {
            System.out.println("Multiplication Table for " + i + ":");
for (int j = 1; j <= 10; j++) {
                System.out.println(i + " x " + j + " = " + (i * j));
            }
            System.out.println();
        }


        scanner.close();
    }
}
```

Output:



```
licationTables }
Enter the number for multiplication tables: 3
Multiplication Table for 1:
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10

Multiplication Table for 2:
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20

Multiplication Table for 3:
3 x 1 = 3
3 x 2 = 6
```

Program 5

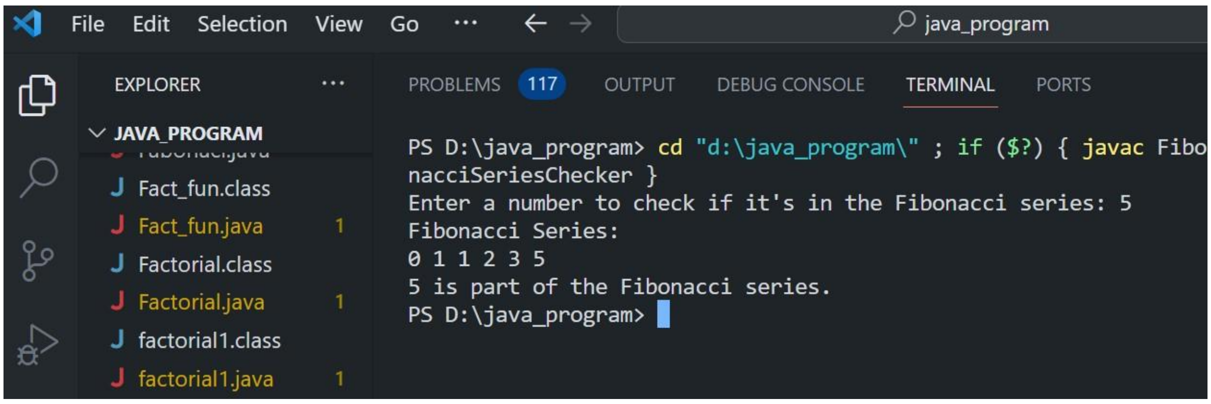
Aim: Write a java program to check input no is part of Fibonacci series or not? Print Fibonacci series till that point.

Code:

```
import java.util.Scanner;

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

        // Accept the input number
        System.out.print("Enter a number to check if it's in the Fibonacci series: ");
int num = scanner.nextInt();

        // Initialize the first two Fibonacci numbers
int a = 0, b = 1;

        // Print the first two Fibonacci numbers
        System.out.println("Fibonacci Series:");
        System.out.print(a + " " + b + " ");

        boolean isPartOfFibonacci = false;

        // Generate and print Fibonacci series until it reaches or exceeds the input number
while (true) {
  int c = a + b;
     if (c > num) {
            break;
        }
```

```
          System.out.print(c + " ");
if (c == num) {
            isPartOfFibonacci = true;
break;
          }
a = b;
b = c;
      }

      if (isPartOfFibonacci) {
          System.out.println("\n" + num + " is part of the Fibonacci series.");
} else {
          System.out.println("\n" + num + " is not part of the Fibonacci series.");
      }

      scanner.close();
   } }
```

Output:



```
PS D:\java_program> cd "d:\java_program\" ; if ($?) { javac Fibo
nacciSeriesChecker }
Enter a number to check if it's in the Fibonacci series: 5
Fibonacci Series:
0 1 1 2 3 5
5 is part of the Fibonacci series.
PS D:\java_program>
```

Program 6

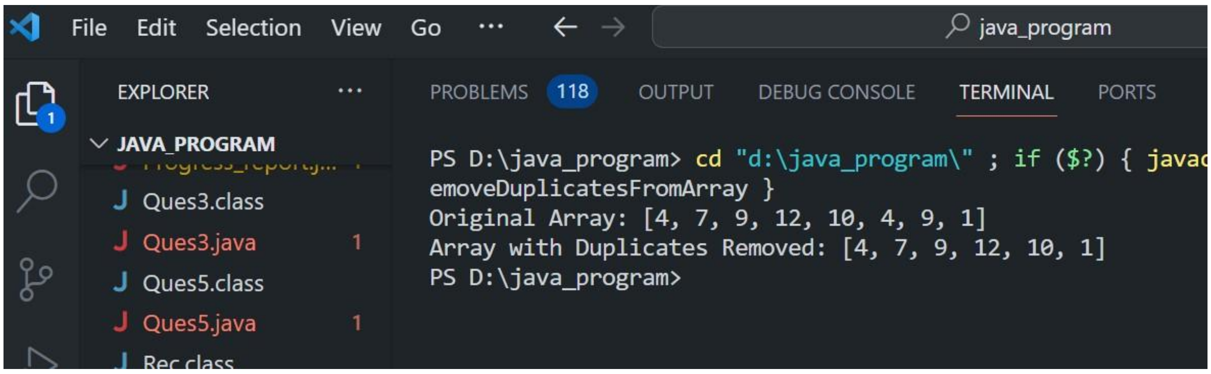
Aim: WAP to remove duplicate elements from the array using a temporary array.

Code:

```
import java.util.Arrays;

public   class   RemoveDuplicatesFromArray   {
public static void main(String[] args) {
   int[] originalArray = {1, 2, 3, 4, 2, 5, 6, 1};

      int[] uniqueArray = removeDuplicates(originalArray);

      System.out.println("Original Array: " + Arrays.toString(originalArray));
      System.out.println("Array with Duplicates Removed: " +
Arrays.toString(uniqueArray));
   }

   public static int[] removeDuplicates(int[] arr) {
int length = arr.length;

      // Create a temporary array to store unique elements
int[] tempArray = new int[length];
  int newSize = 0;

      // Iterate through the original array
for   (int   i   =   0;   i   <   length;   i++)   {
boolean isDuplicate = false;

         // Check if the current element is already in the tempArray
for (int j = 0; j < newSize; j++) {

    if (arr[i] == tempArray[j]) {
```

```
                    isDuplicate = true;
break;
                }
            }

            // If not a duplicate, add it to the tempArray
if (!isDuplicate) {
                tempArray[newSize] = arr[i];
newSize++;
            }
        }

        // Create the final array with unique elements
  int[]  uniqueArray  =  Arrays.copyOf(tempArray,  newSize);
return uniqueArray;
    }
}
```

Output:



```
PS D:\java_program> cd "d:\java_program\" ; if ($?) { javac
emoveDuplicatesFromArray }
Original Array: [4, 7, 9, 12, 10, 4, 9, 1]
Array with Duplicates Removed: [4, 7, 9, 12, 10, 1]
PS D:\java_program>
```

Program 7

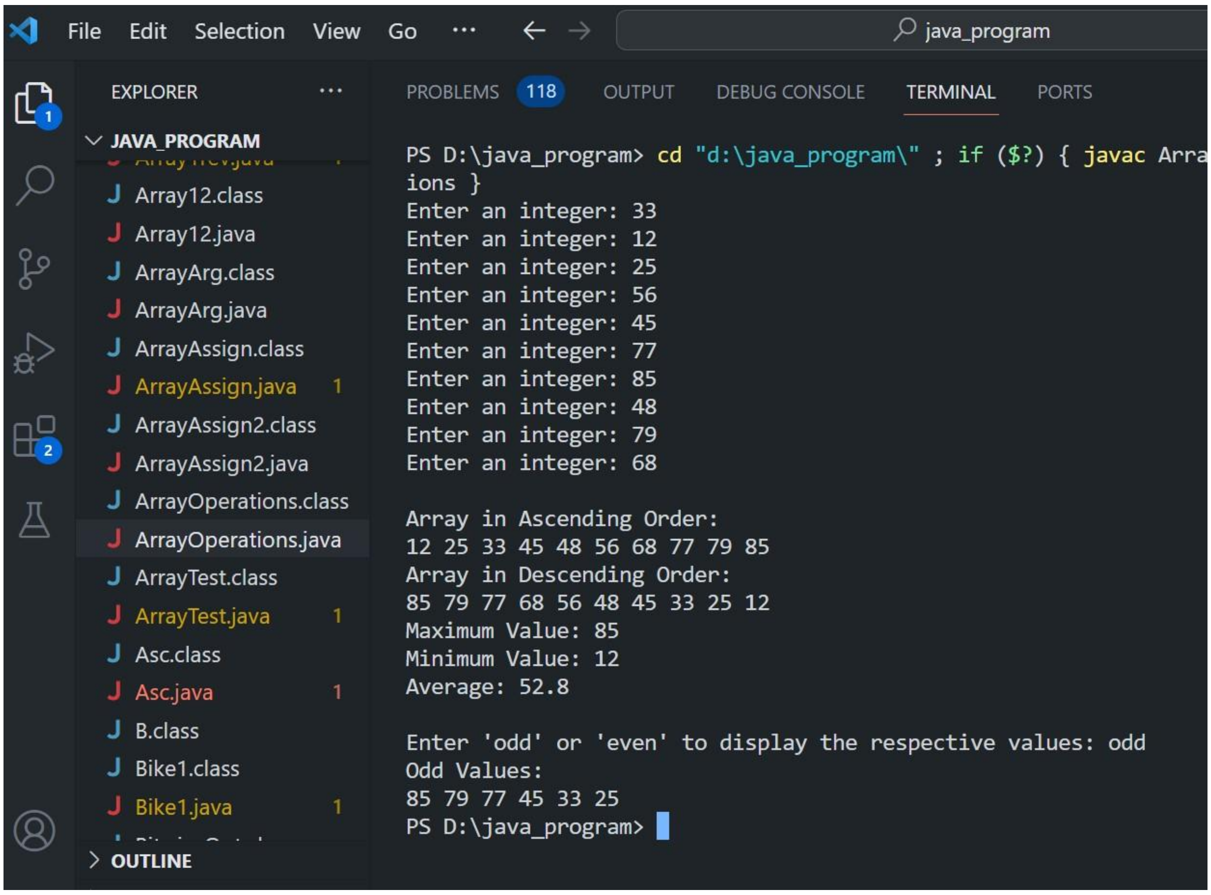
Aim: Write a java program to accept 10 integer values from the user, store them in an array,

i. arrange the array in ascending and descending order, ii.
find the Maximum, minimum and average.
iii. iii. Print only either Odd or Even

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

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

        int[] numbers = new int[10];

        // Accept 10 integer values from the user
for (int i = 0; i < 10; i++) {
            System.out.print("Enter an integer: ");
            numbers[i] = scanner.nextInt();
        }

        // Sort the array in ascending order
        Arrays.sort(numbers);

        // Display the array in ascending order
        System.out.println("\nArray in Ascending Order:");
        for (int num : numbers) {
System.out.print(num + " ");
        }

        // Sort the array in descending order
     for (int i = 0; i < numbers.length / 2; i++) {
      int temp = numbers[i];
  numbers[i]  =  numbers[numbers.length  -  i  -  1];
numbers[numbers.length - i - 1] = temp;
        }

        // Display the array in descending order
        System.out.println("\nArray in Descending Order:");
        for     (int     num     :     numbers)     {
System.out.print(num + " ");        }
```

```
        // Find and display the maximum, minimum, and average
int max = numbers[0];
  int min = numbers[9];
  int sum = 0;
 for (int num : numbers) {
  sum += num;
    if (num > max) {
                max = num;
            }
            if (num < min) {
                min = num;
            }
        }
        double average = (double) sum / numbers.length;

        System.out.println("\nMaximum Value: " + max);
        System.out.println("Minimum Value: " + min);
        System.out.println("Average: " + average);


        System.out.print("\nEnter 'odd' or 'even' to display the respective values: ");
        String choice = scanner.next();

        if (choice.equals("odd")) {
            System.out.println("Odd Values:");
for (int num : numbers) {
     if (num % 2 != 0) {
                    System.out.print(num + " ");
                }
            }
        }   else   if   (choice.equals("even"))   {
System.out.println("EvenValues:");
for (int num : numbers) {
   if (num % 2 == 0) {
                    System.out.print(num + " ");
                }
            }
        } else {
            System.out.println("Invalid choice.");
        }

        scanner.close();
    }
}
```

Output:



```
PS D:\java_program> cd "d:\java_program\" ; if ($?) { javac Arra
ions }
Enter an integer: 33
Enter an integer: 12
Enter an integer: 25
Enter an integer: 56
Enter an integer: 45
Enter an integer: 77
Enter an integer: 85
Enter an integer: 48
Enter an integer: 79
Enter an integer: 68

Array in Ascending Order:
12 25 33 45 48 56 68 77 79 85
Array in Descending Order:
85 79 77 68 56 48 45 33 25 12
Maximum Value: 85
Minimum Value: 12
Average: 52.8

Enter 'odd' or 'even' to display the respective values: odd
Odd Values:
85 79 77 45 33 25
PS D:\java_program>
```

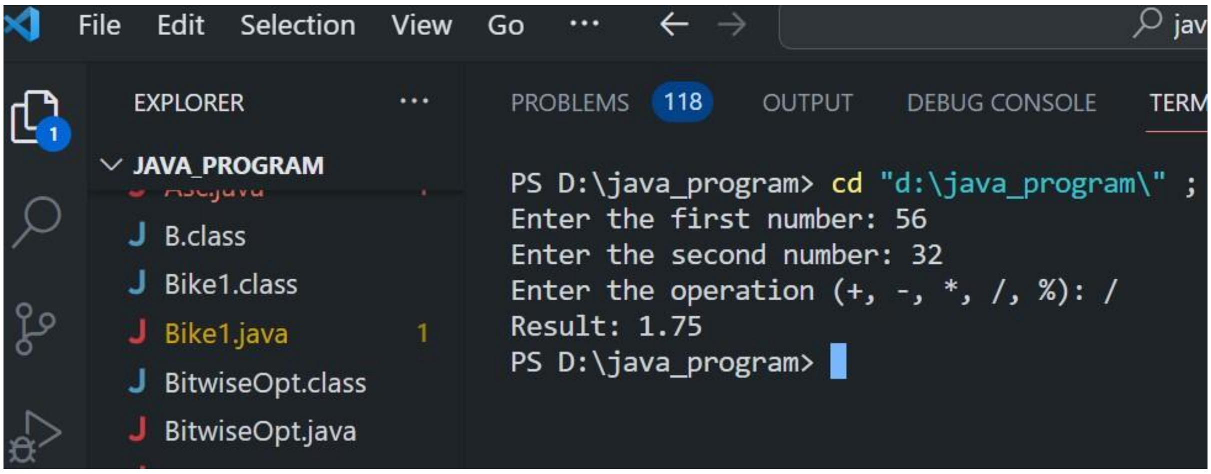
Program 8

Aim: Write a java program to create a calculator. Use classes and methods to perform +,-,*,/,%

Code:

```
import java.util.Scanner;

class Calculator {
   public static double add(double num1, double num2) {
return num1 + num2;
   }

   public static double subtract(double num1, double num2) {
return num1 - num2;
   }

   public static double multiply(double num1, double num2) {
return num1 * num2;
   }

   public static double divide(double num1, double num2) {
if (num2 == 0) {
         System.out.println("Error: Division by zero is not allowed.");
return Double.NaN; // Not-a-Number
      }
      return num1 / num2;
   }

   public static double modulus(double num1, double num2) {
      if (num2 == 0) {
            System.out.println("Error: Modulus by zero is not allowed.");
return Double.NaN; // Not-a-Number
        }
        return num1 % num2;
    }
}

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

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

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

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

        double result = 0;

        switch (operator) {
case '+':
                result = Calculator.add(num1, num2);
break;            case '-':
                result = Calculator.subtract(num1, num2);
break;

            case '*':
                result = Calculator.multiply(num1, num2);
break;          case '/':
                result = Calculator.divide(num1, num2);
break;          case '%':
                result = Calculator.modulus(num1, num2);
break;          default:
                System.out.println("Invalid operator.");
break;
        }

        System.out.println("Result: " + result);

        scanner.close();
    }
}
```

Output:



```
PS D:\java_program> cd "d:\java_program\" ;
Enter the first number: 56
Enter the second number: 32
Enter the operation (+, -, *, /, %): /
Result: 1.75
PS D:\java_program>
```

Program 9

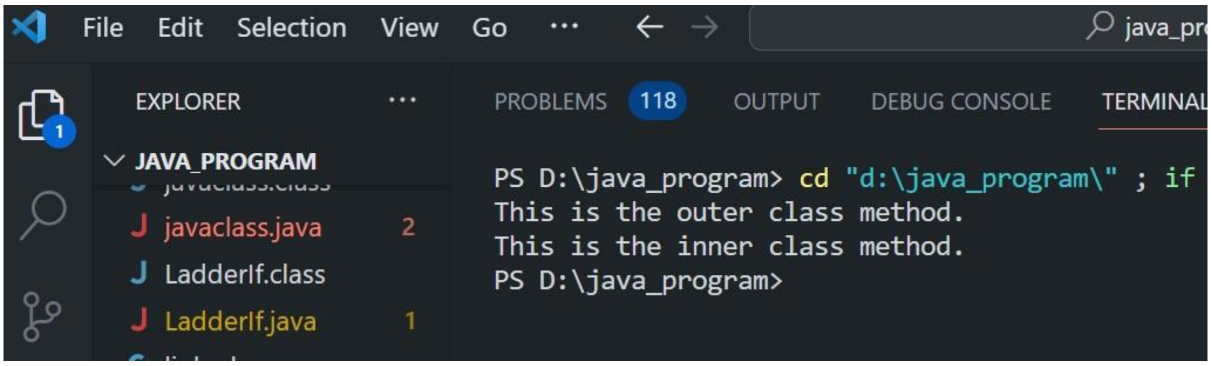
Aim: Create a class “Enclosed” within it create inner class “Nested”, both the classes should have at least one method to display messages. Try to call the method of the “Nested” class in the “Enclosed” class and vice versa.

Code:

```
class Enclosed {
  // Method in the outer class
void outerMethod() {
      System.out.println("This is the outer class method.");
  }

  // Inner class
class Nested {
      // Method in the inner class
void innerMethod() {
         System.out.println("This is the inner class method.");
      }
  }
}

public class Main {
  public static void main(String[] args) {
      // Create an instance of the outer class
      Enclosed outerObj = new Enclosed();

      // Call the outer class method
outerObj.outerMethod();

      // Create an instance of the inner class
      Enclosed.Nested innerObj = outerObj.new Nested();
```

```
        // Call the inner class method from the outer class
innerObj.innerMethod();
    }
}
```

Output:



```
PS D:\java_program> cd "d:\java_program\" ; if
This is the outer class method.
This is the inner class method.
PS D:\java_program>
```

Program 10

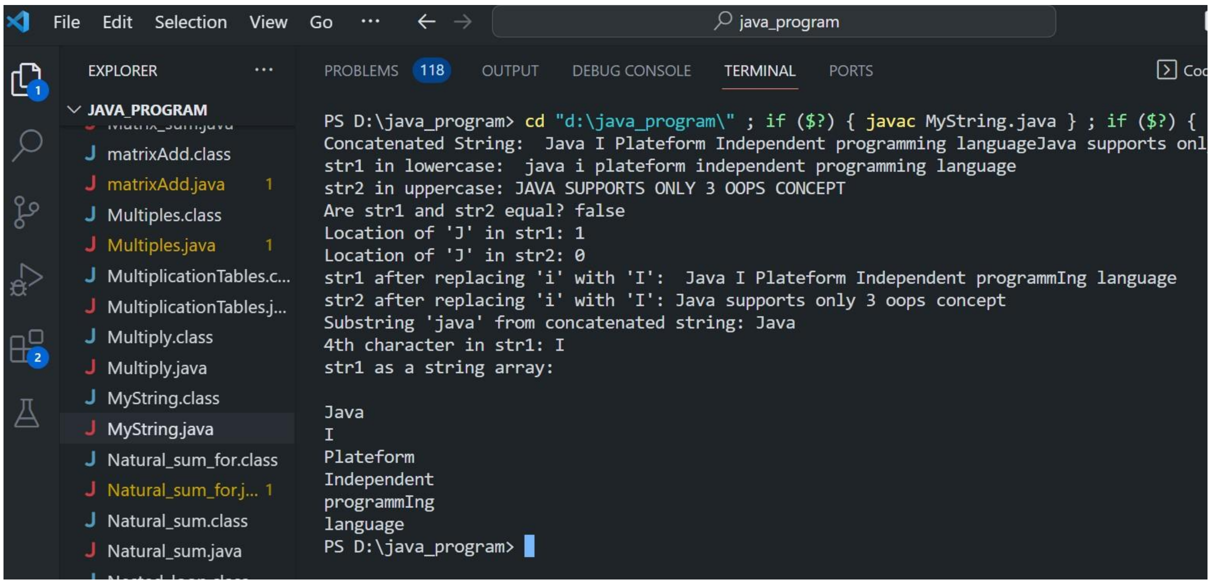
Aim: Create a class called MyString : Declare two string type variables: str1 (“ Welcome to Java tutorial”) and str2(“Today's topic is String Handling in Java”). Perform following operations in this class:

i. Concatenate two strings
ii. . Covert str1 into lower case
iii. Covert str2 into upper case
iv. Are both equal to each other
v. Show the location of “J” in both str1 and str2
vi. Replace “i” with “I” in both the strings
vii. display “java” from str string
Vii Display the “7” character in str1.
ix. Convert str1 into string array

Code:

```
public class MyString {
  public static void main(String[] args) {
        // Declare two string variables
        String str1 = " Welcome to Java tutorial";
        String str2 = "Today's topic is String Handling in Java";

        // Concatenate two strings
        String concatenatedString = str1 + str2;
        System.out.println("Concatenated String: " + concatenatedString);

        // Convert str1 to lowercase
        String str1LowerCase = str1.toLowerCase();
        System.out.println("str1 in lowercase: " + str1LowerCase);

        // Convert str2 to uppercase
        String str2UpperCase = str2.toUpperCase();
        System.out.println("str2 in uppercase: " + str2UpperCase);

        // Check if both strings are equal
boolean areEqual = str1.equals(str2);
        System.out.println("Are str1 and str2 equal? " + areEqual);

        // Find the location of "J" in both strings
int indexInStr1 = str1.indexOf("J");
   int indexInStr2 = str2.indexOf("J");
System.out.println("Location of 'J' in str1: " +
indexInStr1);
System.out.println("Location of 'J' in str2: " +
indexInStr2);

        // Replace "i" with "I" in both strings
str1 = str1.replace("i", "I");
  str2 = str2.replace("i", "I");
        System.out.println("str1 after replacing 'i' with 'I': " + str1);
System.out.println("str2 after replacing 'i' with 'I': " + str2);

        // Display "java" from str string
        String javaSubstring =
concatenatedString.substring(concatenatedString.indexOf("Java"),
concatenatedString.indexOf("Java") + 4);
        System.out.println("Substring 'java' from concatenated string: " + javaSubstring);

        // Display the 7th character in str1
char seventhChar = str1.charAt(6);
        System.out.println("7th character in str1: " + seventhChar);

        // Convert str1 into a string array
        String[] str1Array = str1.split(" ");
        System.out.println("str1 as a string array: ");
for (String word : str1Array) {
            System.out.println(word);
        }
    }
}
```

Output:



```
PS D:\java_program> cd "d:\java_program\" ; if ($?) { javac MyString.java } ; if ($?) {
Concatenated String:  Java I Plateform Independent programming languageJava supports onl
str1 in lowercase:  java i plateform independent programming language
str2 in uppercase: JAVA SUPPORTS ONLY 3 OOPS CONCEPT
Are str1 and str2 equal? false
Location of 'J' in str1: 1
Location of 'J' in str2: 0
str1 after replacing 'i' with 'I':  Java I Plateform Independent programmIng language
str2 after replacing 'i' with 'I': Java supports only 3 oops concept
Substring 'java' from concatenated string: Java
4th character in str1: I
str1 as a string array:

Java
I
Plateform
Independent
programmIng
language
PS D:\java_program>
```

Program 11

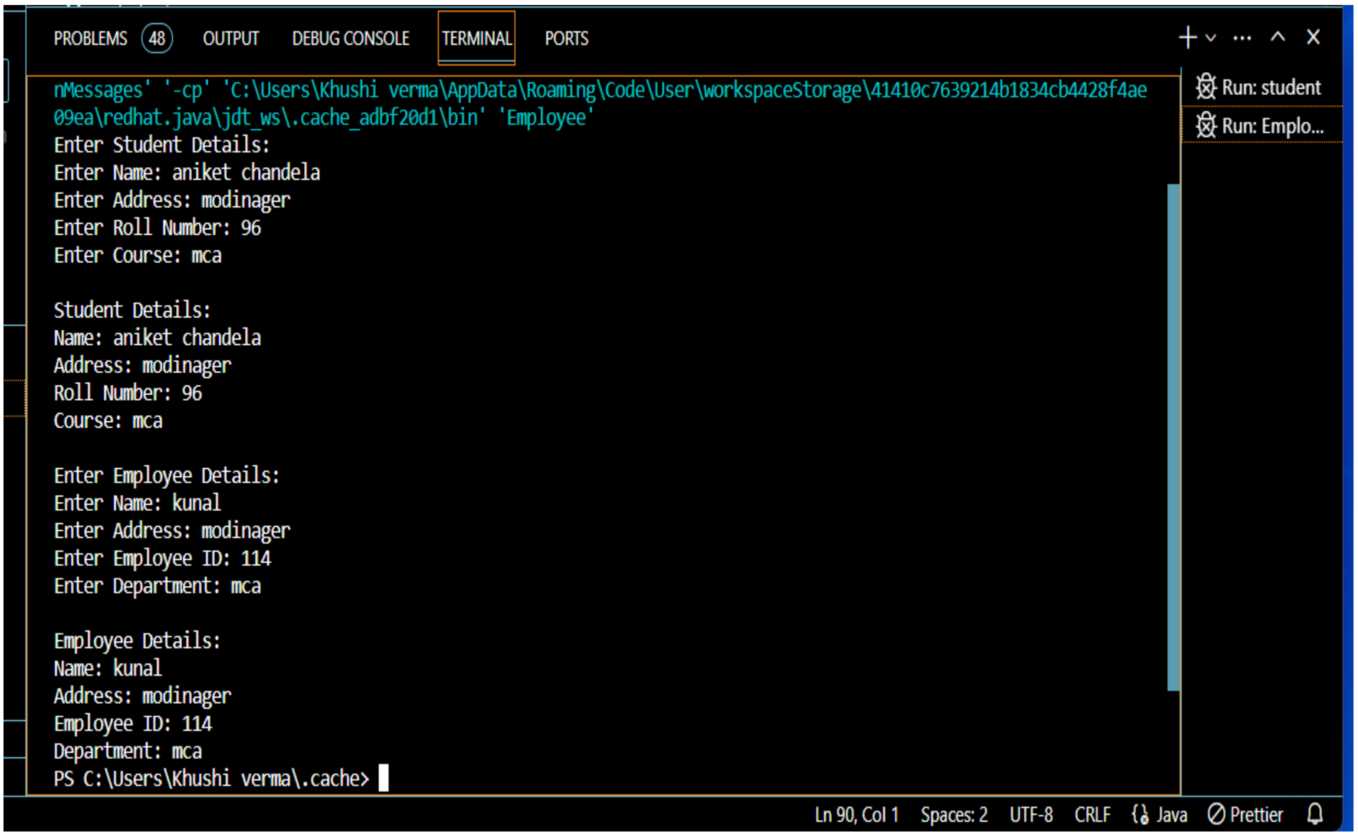
Aim: Create a class person (Data Member: Name & address , Method: Accept() and display() to accept and display value of data member on Output device. Derive two classes student (Data Member: Rollno, Course Member Method: Accept() and Display()) and Employee((Data Member: EmpId ,Department Member Method: Accept() and Display()).display details of one student and one employee. NOTE: use super keyword to invoke hidden members of base class.

Code:

```
import java.util.Scanner;


class Person {
  protected String name;
protected String address;


   // Constructor for Person class
public Person() {
    name = "";
     address = "";
   }


   // Method to accept person details
public void Accept() {
       Scanner scanner = new Scanner(System.in);
System.out.print("Enter Name: ");
     name = scanner.nextLine();
System.out.print("Enter Address: ");
address = scanner.nextLine();
   }
   // Method to display person details
   public void Display() {
       System.out.println("Name: " + name);
       System.out.println("Address: " + address);
    }
}

class Student extends Person {
private int rollNo;
   private String course;

    // Constructor for Student class
public Student() {
        super(); // Invoke base class constructor
rollNo = 0;
 course = "";
    }

    // Method to accept student details
  public void Accept() {
  super.Accept(); // Invoke base class method
        Scanner scanner = new Scanner(System.in);
System.out.print("Enter Roll Number: ");
rollNo = scanner.nextInt();
scanner.nextLine(); // Consume newline
System.out.print("Enter Course: ");
     course = scanner.nextLine();
    }

    // Method to display student details public
    void Display() {

  super.Display(); // Invoke base class
    method
```

```
        System.out.println("Roll Number: " + rollNo);
        System.out.println("Course: " + course);
    }
}

class Employee extends Person {
private int empId;
   private String department;

    // Constructor for Employee class
    public Employee() {
  super();  //  Invoke  base  class  constructor
empId = 0;
department = "";
    }

    // Method to accept employee details
 public void Accept() {
    super.Accept(); // Invoke base class method
        Scanner scanner = new Scanner(System.in);
System.out.print("Enter Employee ID: ");
empId = scanner.nextInt();
     scanner.nextLine(); // Consume newline
System.out.print("Enter Department: ");
department = scanner.nextLine();
    }

    // Method to display employee details public
    void Display() {
        super.Display(); // Invoke base class method
        System.out.println("Employee ID: " + empId);
        System.out.println("Department: " + department);
    }
}

public class Main {
    public static void main(String[] args) {
        Student student = new Student();
System.out.println("Enter Student Details:");
student.Accept();
        System.out.println("\nStudent Details:");
student.Display();

        Employee employee = new Employee();
System.out.println("\nEnter Employee Details:");
employee.Accept();
        System.out.println("\nEmployee Details:");
employee.Display();
    }
}
```

Output:



```
nMessages' '-cp' 'C:\Users\Khushi verma\AppData\Roaming\Code\User\workspaceStorage\41410c7639214b1834cb4428f4ae
09ea\redhat.java\jdt_ws\.cache_adbf20d1\bin' 'Employee'
Enter Student Details:
Enter Name: aniket chandela
Enter Address: modinager
Enter Roll Number: 96
Enter Course: mca

Student Details:
Name: aniket chandela
Address: modinager
Roll Number: 96
Course: mca

Enter Employee Details:
Enter Name: kunal
Enter Address: modinager
Enter Employee ID: 114
Enter Department: mca

Employee Details:
Name: kunal
Address: modinager
Employee ID: 114
Department: mca
PS C:\Users\Khushi verma\.cache>
```

`

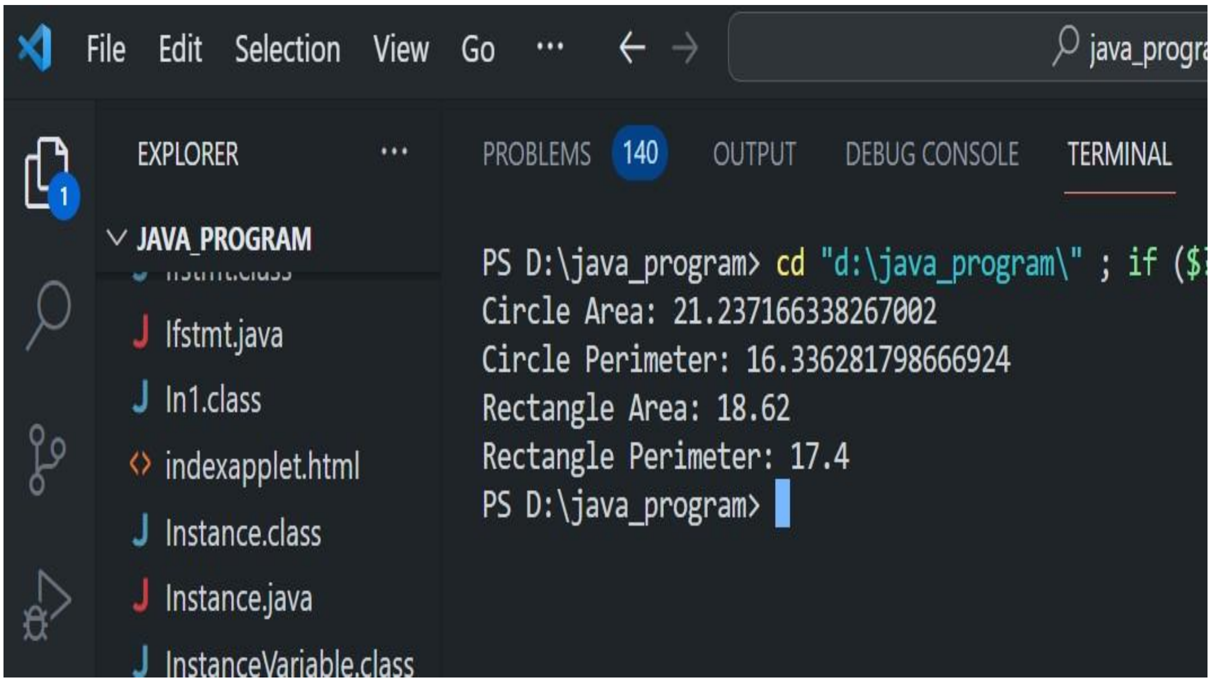
Program 12

Aim: WAP to show the use of Interfaces in java

Code:

```
interface Shape {
   double calculateArea(); // Abstract method (method without a body)
double calculatePerimeter(); // Another abstract method
}

// Implement the "Shape" interface in a class
class Circle implements Shape {
 private double radius;

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

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

   @Override
   public double calculatePerimeter() {
return 2 * Math.PI * radius;
   }
}

// Implement the "Shape" interface in another class
class Rectangle implements Shape {
 private double length;
 private double width;

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

   @Override
   public double calculateArea() {
return length * width;
   }

   @Override
   public double calculatePerimeter() {
return 2 * (length + width);
   }
}

public class Main {
   public static void main(String[] args) {
       Circle circle = new Circle(5.0);
       Rectangle rectangle = new Rectangle(4.0, 6.0);

       // Calculate and display the area and perimeter of shapes
       System.out.println("Circle Area: " + circle.calculateArea());
       System.out.println("Circle Perimeter: " + circle.calculatePerimeter());
System.out.println("Rectangle Area: " + rectangle.calculateArea());
       System.out.println("Rectangle Perimeter: " + rectangle.calculatePerimeter());
   }
}
```

Output:



```
PS D:\java_program> cd "d:\java_program\" ; if ($
Circle Area: 21.237166338267002
Circle Perimeter: 16.336281798666924
Rectangle Area: 18.62
Rectangle Perimeter: 17.4
PS D:\java_program>
```

Program 13

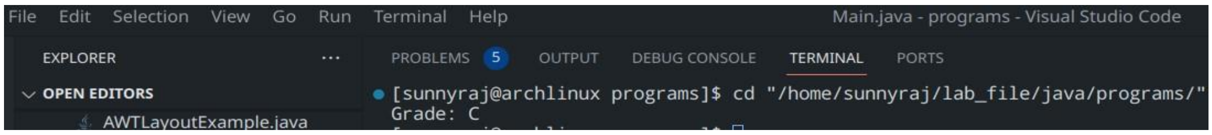
Aim: Write a java program to display the grade of students depending on marks, please raise a user defined checked Exception, if less than 0 or more than 100 marks are entered for grade

Code:

```
// Define a custom checked exception class class
InvalidMarksException extends Exception {
public InvalidMarksException(String message) {
super(message);
   }
}

// Create a class for student grading class
StudentGrading {
   public static char calculateGrade(int marks) throws InvalidMarksException {
if (marks < 0 || marks > 100) {
          throw new InvalidMarksException("Invalid marks: Marks should be between 0 and
100.");
       }

       if (marks >= 90) {
return 'A';
       } else if (marks >= 80) {
return 'B';
       } else if (marks >= 70) {
return 'C';
       } else if (marks >= 60) {
return 'D';
       } else {
          return 'F';
       }
```

```
    }
}

public class Main {
    public static void main(String[] args) {
        try {
            int studentMarks = 75; // Change this to test different marks
char grade = StudentGrading.calculateGrade(studentMarks);
            System.out.println("Grade: " + grade);
        } catch (InvalidMarksException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Output:



File Edit Selection View Go Run Terminal Help
Main.java - programs - Visual Studio Code
EXPLORER
OPEN EDITORS
AWTLayoutExample.java
PROBLEMS 5
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```
[sunnyraj@archlinux programs]$ cd "/home/sunnyraj/lab_file/java/programs/"
Grade: C
```

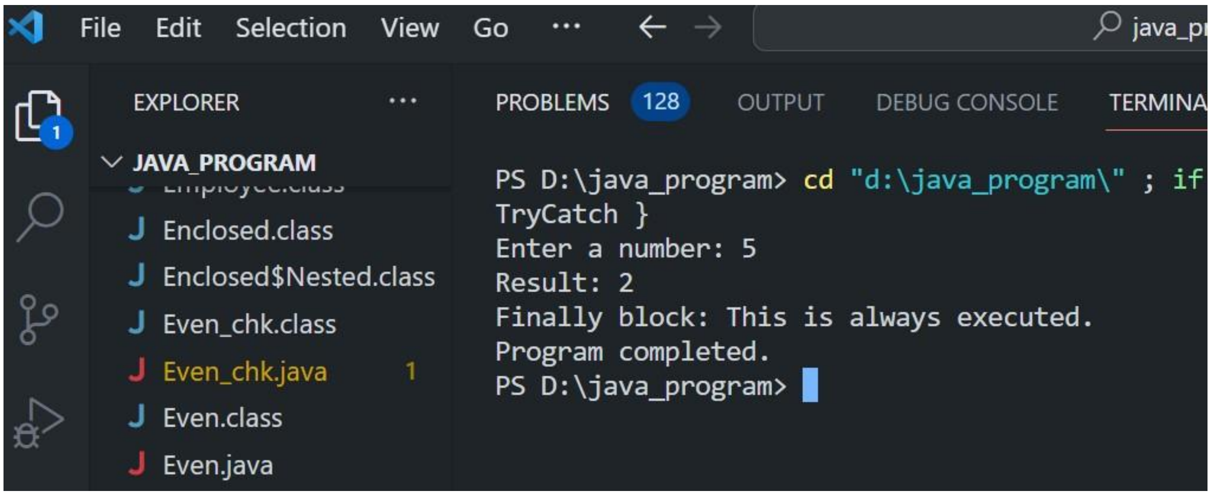
Program 14

Aim: Write a Java program to use the try and catch and finally block.

Code:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
         try
{
            System.out.print("Enter a number: ");
int number = scanner.nextInt();
            int result = 10 / number; // This may cause an ArithmeticException

            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Error: Division by zero or other arithmetic error.");
        } catch (java.util.InputMismatchException e) {
            System.out.println("Error: Invalid input. Please enter a valid number.");
} finally {
            // This block is always executed, regardless of whether an exception occurred or not
System.out.println("Finally block: This is always executed.");
   scanner.close();
        }

        System.out.println("Program completed.");
    }
}
```

Output:



```
PS D:\java_program> cd "d:\java_program\" ; if
TryCatch }
Enter a number: 5
Result: 2
Finally block: This is always executed.
Program completed.
PS D:\java_program>
```

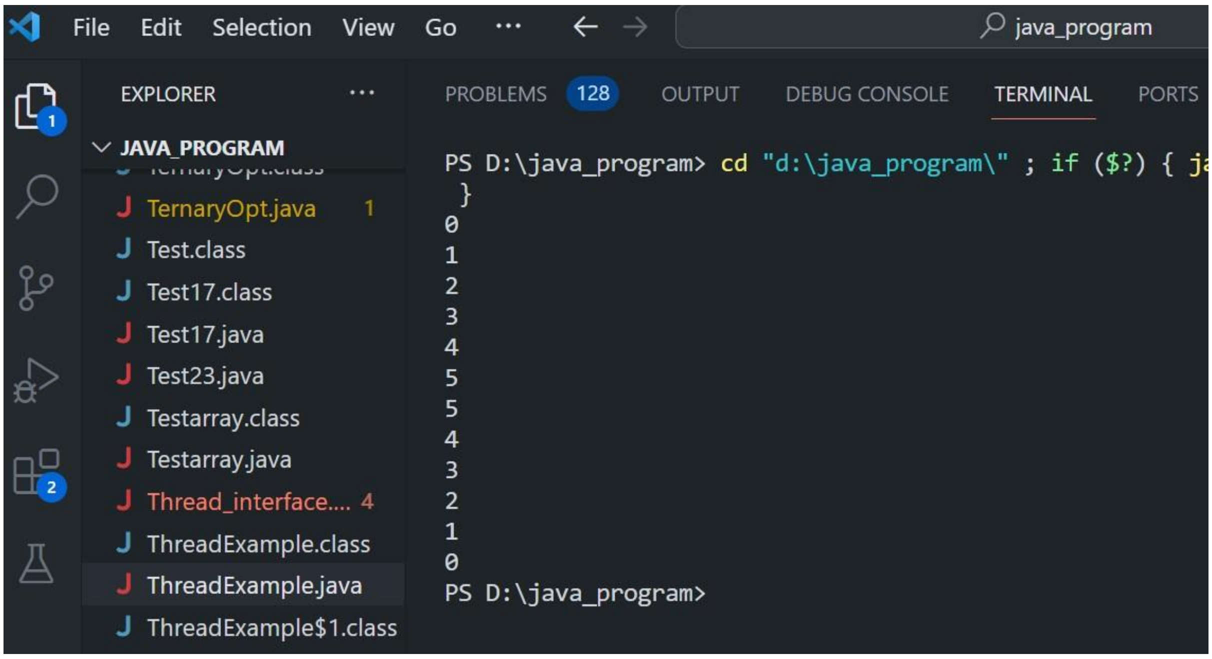
Program 15

Aim: Write a multithreaded program where one thread will print 0-5 and another thread will print 5-0. Use thread class.

Code:

```
public class ThreadExample {
 public static void main(String[] args) {
       Thread thread1 = new Thread(new Runnable() {
public void run() {
    for (int i = 0; i <= 5; i++) {
                System.out.println(i);
            }
         }
       });

       Thread thread2 = new Thread(new Runnable() {
public void run() {
      for (int i = 5; i >= 0; i--) {
                System.out.println(i);
            }
         }
       });

       thread1.start();
thread2.start();
   }
}
```

Output:



```
PS D:\java_program> cd "d:\java_program\" ; if ($?) { ja
 }
0
1
2
3
4
5
5
4
3
2
1
0
PS D:\java_program>
```

Program 16

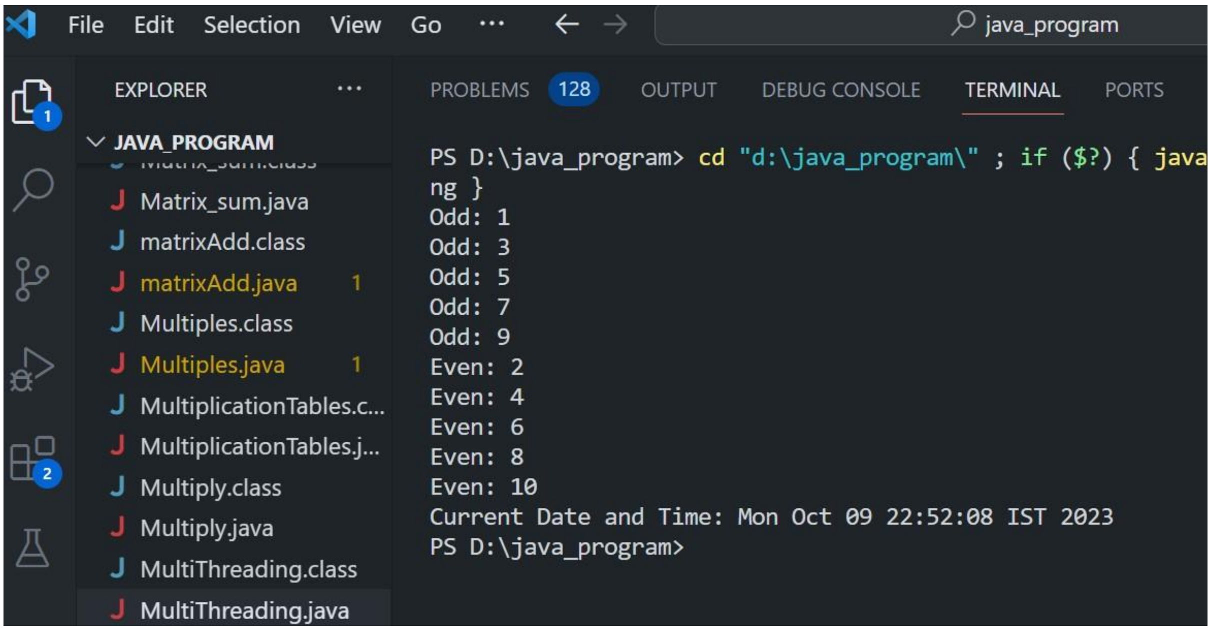
Aim: Write a java multithreaded (2 or more)java program , one thread will print odd numbers and another will print even numbers and the main thread is there it will print date and time. Use Runnable interface.

Code:

```
import java.util.Date;

class EvenNumberRunnable implements Runnable {
public void run() {
      for (int i = 2; i <= 10; i += 2) {
         System.out.println("Even: " + i);
      }
   }
}

class OddNumberRunnable implements Runnable {
public void run() {
      for (int i = 1; i <= 9; i += 2) {
         System.out.println("Odd: " + i);
      }
   }
}

public class MultiThreadExample {
public static void main(String[] args) {
      Runnable evenTask = new EvenNumberRunnable();
      Runnable oddTask = new OddNumberRunnable();

      Thread evenThread = new Thread(evenTask);
```

```
        Thread oddThread = new Thread(oddTask);

        evenThread.start();
    oddThread.start();

        Date currentDate = new Date();
        System.out.println("Current Date and Time: " + currentDate);
      }
    }
```

Output:



```
PS D:\java_program> cd "d:\java_program\" ; if ($?) { java
ng }
Odd: 1
Odd: 3
Odd: 5
Odd: 7
Odd: 9
Even: 2
Even: 4
Even: 6
Even: 8
Even: 10
Current Date and Time: Mon Oct 09 22:52:08 IST 2023
PS D:\java_program>
```

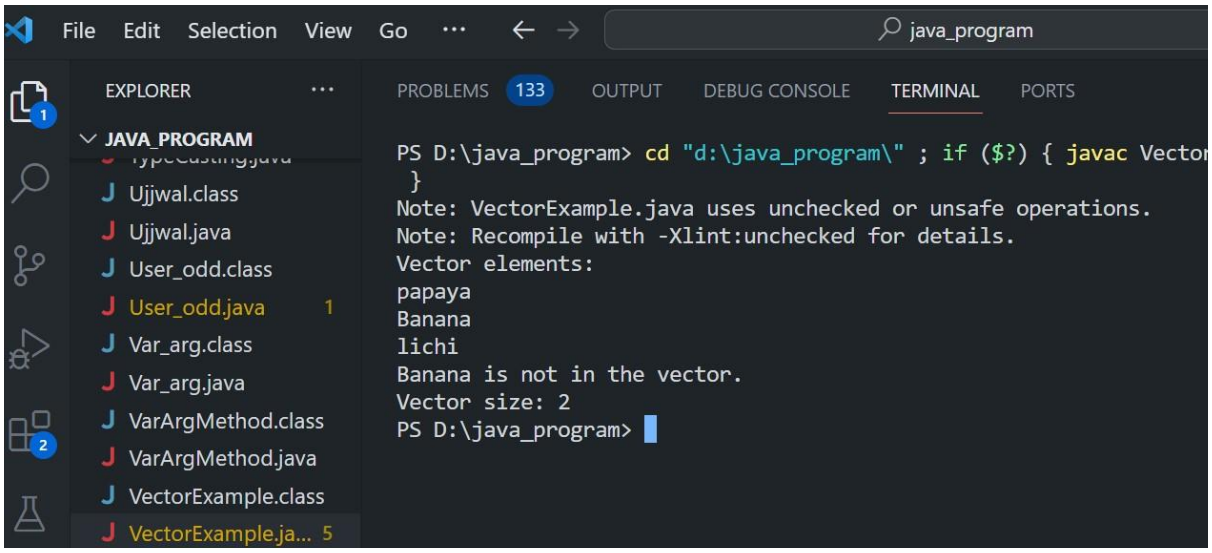
Program 17

Aim: WAP to show the use of Legacy classes:- Vector

Code:

```
import java.util.Vector;

public class VectorExample {
  public static void main(String[] args) {
        Vector vector = new Vector();

        vector.add("Apple");
vector.add("Banana");
vector.add("Cherry");

        System.out.println("Vector elements:");
for (Object fruit : vector) {
            System.out.println(fruit);
        }

        vector.remove("Banana");

        if (vector.contains("Banana")) {
            System.out.println("Banana is in the vector.");
        } else {
            System.out.println("Banana is not in the vector.");
        }

        System.out.println("Vector size: " + vector.size());
    }
}
```

Output:



```
PS D:\java_program> cd "d:\java_program\" ; if ($?) { javac Vector
 }
Note: VectorExample.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Vector elements:
papaya
Banana
lichi
Banana is not in the vector.
Vector size: 2
PS D:\java_program>
```

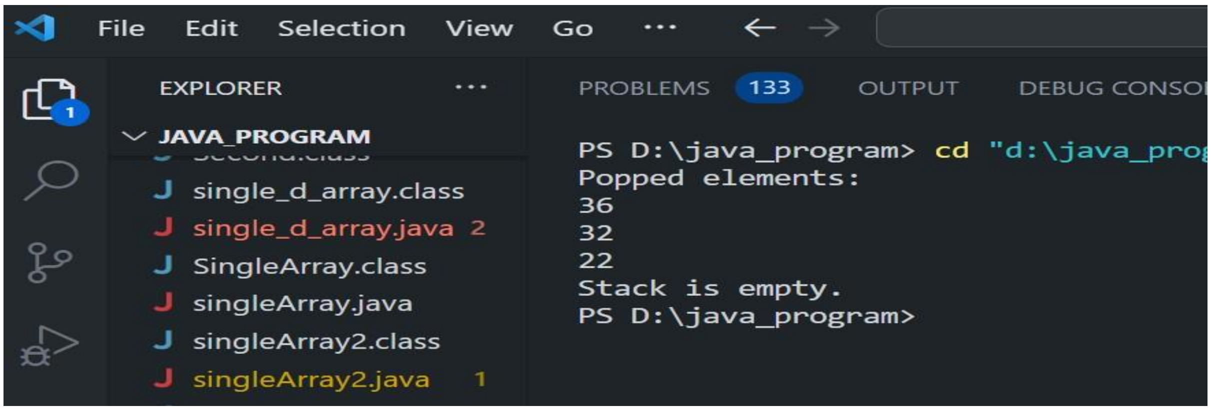
Program 18

Aim: WAP to show the use of Legacy classes:- Stack

Code:

```
import java.util.Stack;


public class StackExample {
  public static void main(String[] args) {
        Stack<Integer> stack = new Stack<>();


        stack.push(1);
stack.push(2);
  stack.push(3);


        System.out.println("Popped elements:");
while (!stack.isEmpty()) {
            System.out.println(stack.pop());
        }
        if (stack.isEmpty()) {
            System.out.println("Stack is empty.");
        }
    }
}
```

Output:



```
PS D:\java_program> cd "d:\java_prog
Popped elements:
36
32
22
Stack is empty.
PS D:\java_program>
```

Program 19

Aim: WAP to demonstrate the use of followings:

i. StringTokenizer

ii. Date iii.

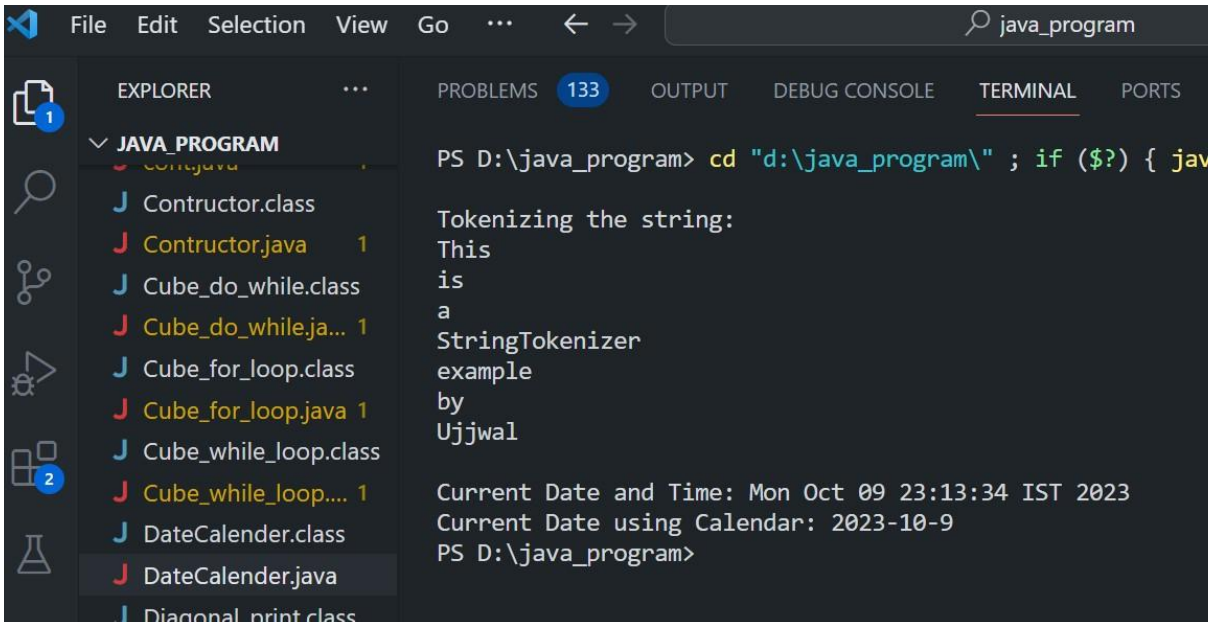
Calendar

Code:

```
import java.util.StringTokenizer;
import java.util.Date; import
java.util.Calendar;

public class STDateCalendarExample {
public static void main(String[] args) {
    // i. Using StringTokenizer
    String text = "This is a StringTokenizer example";
    StringTokenizer tokenizer = new StringTokenizer(text);

    System.out.println("Tokenizing the string:");
while (tokenizer.hasMoreTokens()) {
        System.out.println(tokenizer.nextToken());
    }

    // ii. Using Date
    Date currentDate = new Date();
    System.out.println("\nCurrent Date and Time: " + currentDate);

    // iii. Using Calendar
    Calendar calendar = Calendar.getInstance();
int year = calendar.get(Calendar.YEAR);
```

```
        int month = calendar.get(Calendar.MONTH) + 1; // Months are 0-based
int day = calendar.get(Calendar.DAY_OF_MONTH);

        System.out.println("Current Date using Calendar: " + year + "-" + month + "-" + day);
    }
}
```

Output:



```
PS D:\java_program> cd "d:\java_program\" ; if ($?) { jav

Tokenizing the string:
This
is
a
StringTokenizer
example
by
Ujjwal

Current Date and Time: Mon Oct 09 23:13:34 IST 2023
Current Date using Calendar: 2023-10-9
PS D:\java_program>
```

Program 20

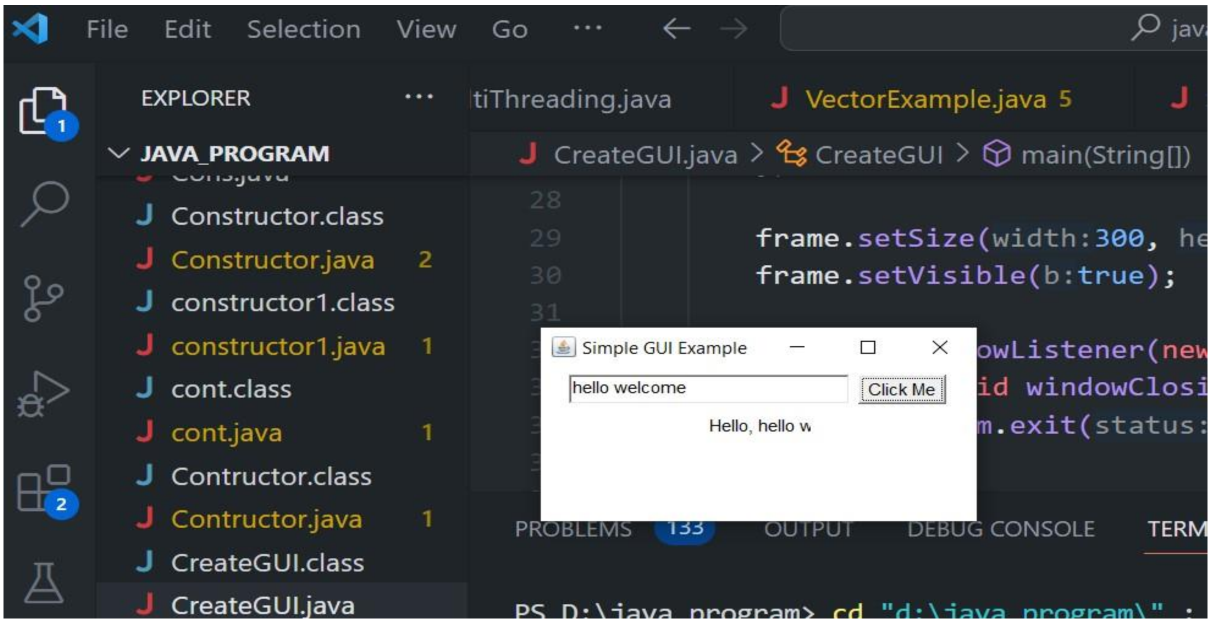
Aim: WAP to create a Simple GUI with text field button and label and handle click event of button.

Code:

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

public class SimpleGUIExample {
private Frame frame;
   private TextField textField;
private Button button;
  private Label label;

   public SimpleGUIExample() {
       frame = new Frame("Simple GUI Example");

       textField = new TextField(20);
button = new Button("Click Me");
label = new Label("Welcome!");

       frame.setLayout(new FlowLayout());
frame.add(textField);
frame.add(button);
    frame.add(label);

       button.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
               String text = textField.getText();
               label.setText("Hello, " + text + "!");
```

```
            }
        });

        frame.setSize(300, 150);
    frame.setVisible(true);

        frame.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }

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

Output:



```
frame.setSize(width:300, he
frame.setVisible(b:true);
owListener(new
id windowClosi
m.exit(status:
```

```
hello welcome
Hello, hello w
```

```
PS D:\java program> cd "d:\java program\" ;
```

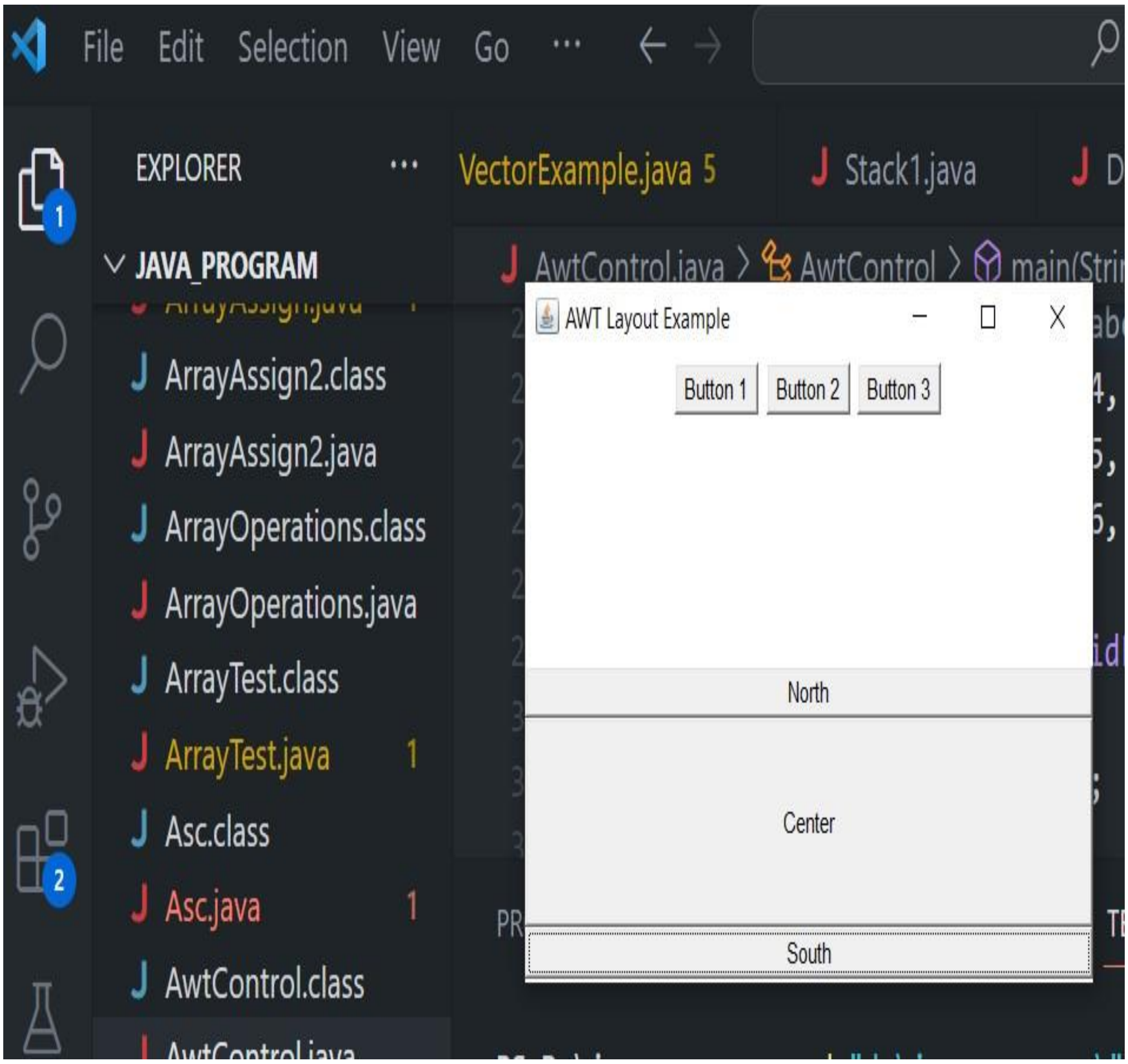
Program 21

Aim: WAP to show different layouts using AWT controls.

Code:

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

public class AWTLayoutExample {
private Frame frame;
    private Button button1, button2, button3, button4, button5, button6;

    public AWTLayoutExample() {
   frame = new Frame("AWT Layout Example");

        // FlowLayout
        Panel flowPanel = new Panel(new FlowLayout());
button1 = new Button("Button 1");
      button2 = new Button("Button 2");
   button3 = new Button("Button 3");
flowPanel.add(button1);
  flowPanel.add(button2);
flowPanel.add(button3);

        // BorderLayout
        Panel borderPanel = new Panel(new BorderLayout());
button4 = new Button("North");
  button5 = new Button("Center");
    button6 = new Button("South");
borderPanel.add(button4, BorderLayout.NORTH);
borderPanel.add(button5, BorderLayout.CENTER);
borderPanel.add(button6, BorderLayout.SOUTH);
frame.setLayout(new GridLayout(2, 1));
frame.add(flowPanel);
 frame.add(borderPanel);

        frame.setSize(400, 300);
frame.setVisible(true);

        frame.addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }

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

Output:


File
Edit
Selection
View
Go
EXPLORER
JAVA_PROGRAM
ArrayAssign2.class
ArrayAssign2.java
ArrayOperations.class
ArrayOperations.java
ArrayTest.class
ArrayTest.java
Asc.class
Asc.java
AwtControl.class
VectorExample.java
Stack1.java
AwtControl.java > AwtControl > main(Stri
AWT Layout Example
Button 1
Button 2
Button 3
North
Center
South

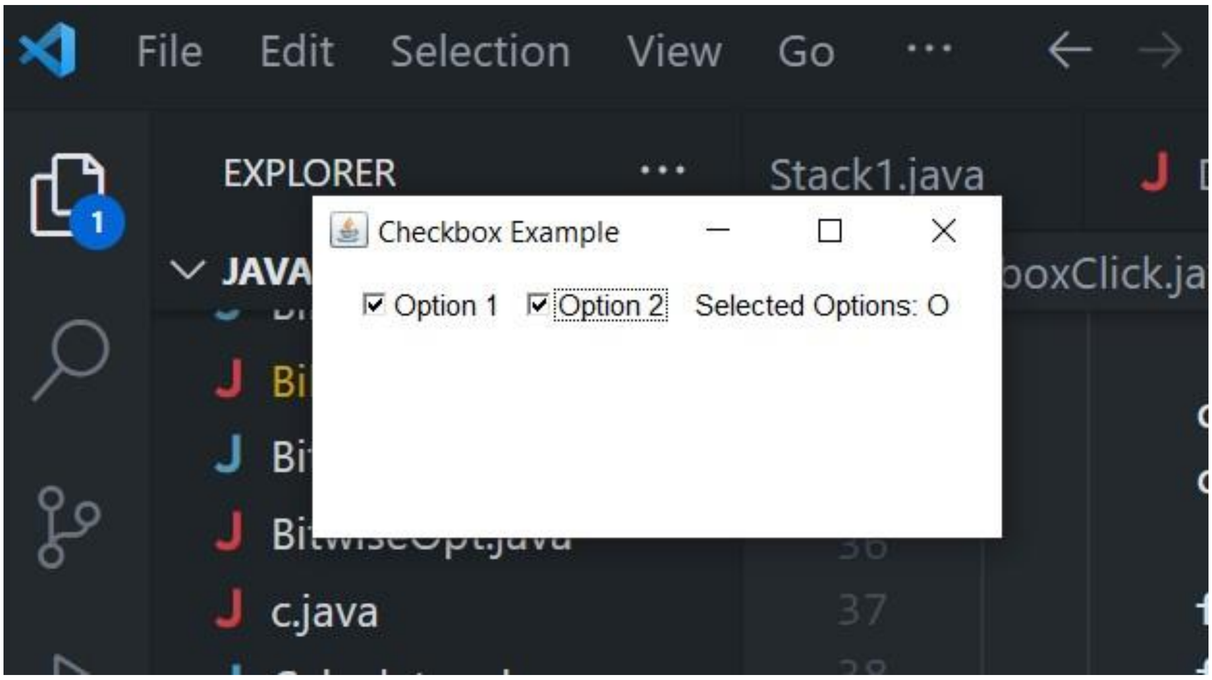
Program 22

Aim: WAP to create a GUI to show checkboxes handling their events.

Code:

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

public class CheckboxExample {
private Frame frame;
 private Checkbox checkBox1;
private Checkbox checkBox2;
private Label label;

   public CheckboxExample() {
   frame = new Frame("Checkbox Example");
checkBox1 = new Checkbox("Option 1");
checkBox2 = new Checkbox("Option 2");
label = new Label("Selected Options:");

       frame.setLayout(new FlowLayout());
frame.add(checkBox1);
frame.add(checkBox2);
frame.add(label);

       ItemListener itemListener = new ItemListener() {
public    void    itemStateChanged(ItemEvent    e)    {
String selectedOptions = "";
   if (checkBox1.getState()) {
               selectedOptions += checkBox1.getLabel() + " ";
            }
            if (checkBox2.getState()) {
                    selectedOptions += checkBox2.getLabel() + " ";
                }
                label.setText("Selected Options: " + selectedOptions);
            }
        };

        checkBox1.addItemListener(itemListener);
checkBox2.addItemListener(itemListener);

        frame.setSize(300, 150);
frame.setVisible(true);
        frame.addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }

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

Output:



```
☑ Option 1   ☑ Option 2   Selected Options: O
```

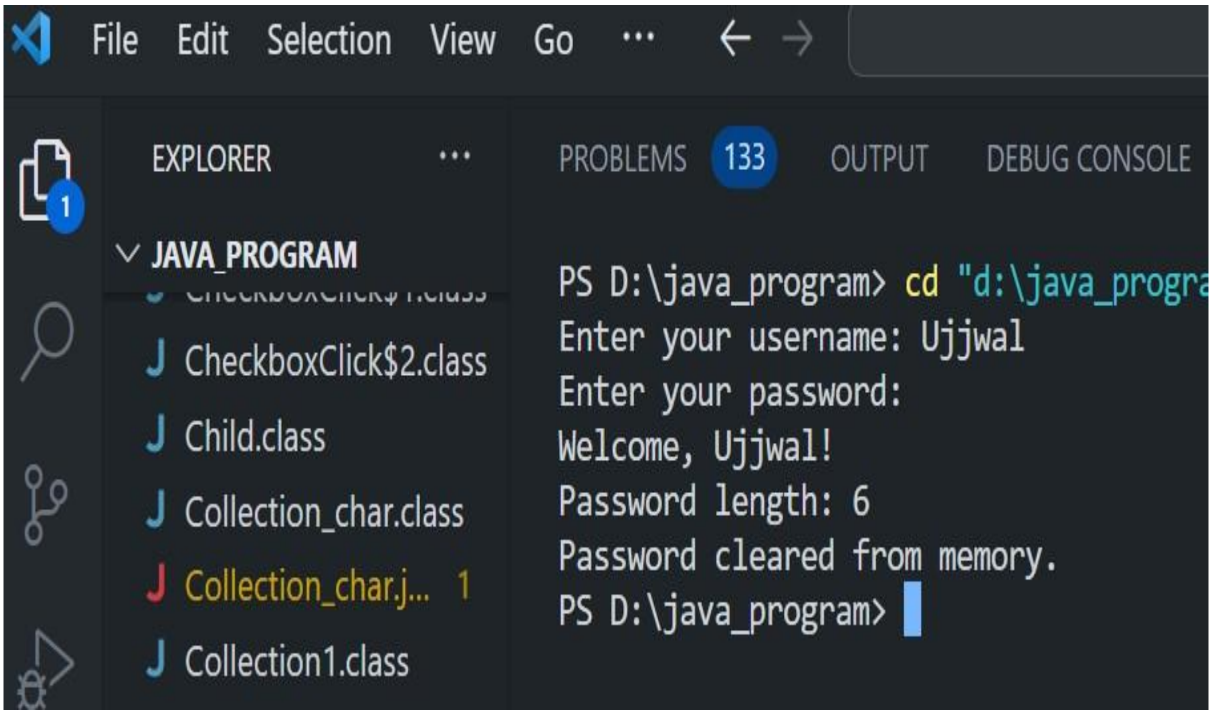
Program 23

Aim: WAP to show the use of Console class for reading and writing.

Code:

```
import java.io.Console;

public class ConsoleExample {
  public static void main(String[] args) {
        Console console = System.console();

        if (console == null) {
            System.out.println("Console is not available.");
            System.exit(1);
        }

        String username = console.readLine("Enter your username: ");
char[] passwordArray = console.readPassword("Enter your password: ");

        console.printf("Welcome, %s!\n", username);
        console.printf("Password length: %d\n", passwordArray.length);

        for (int i = 0; i < passwordArray.length; i++) {
passwordArray[i] = ' ';
        }

        console.printf("Password cleared from memory.\n");
    }
}
```

Output:



```
PS D:\java_program> cd "d:\java_progra
Enter your username: Ujjwal
Enter your password:
Welcome, Ujjwal!
Password length: 6
Password cleared from memory.
PS D:\java_program>
```

Program 24

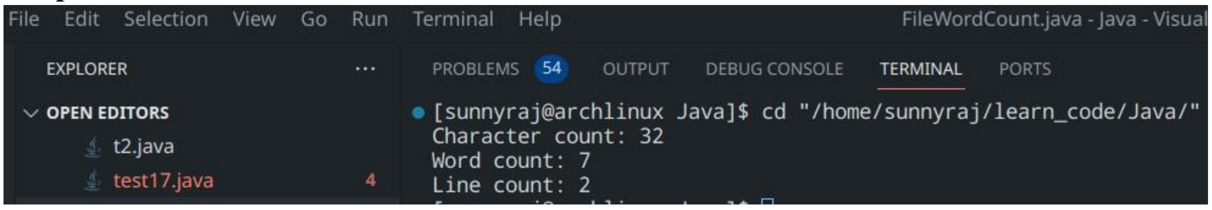
Aim: WAP to count the number of characters, words and lines in a file.

Code:

```
import java.io.*;

public class FileWordCount {     public
static void main(String[] args) {

        String filename = "file.txt";
int charCount = 0;
  int wordCount = 0;
 int lineCount = 0;

        try (BufferedReader reader = new BufferedReader(new FileReader(filename))) {
            String line;
            while ((line = reader.readLine()) != null) {
charCount += line.length();
                String[] words = line.trim().split("\\s+");
wordCount += words.length;
  lineCount++;
            }
        } catch (IOException e) {
            System.err.println("Error reading the file: " + e.getMessage());
            System.exit(1);
        }

        System.out.println("Character count: " + charCount);
        System.out.println("Word count: " + wordCount);
        System.out.println("Line count: " + lineCount);
```

```
    }
}
```

Output:



```
[sunnyraj@archlinux Java]$ cd "/home/sunnyraj/learn_code/Java/"
Character count: 32
Word count: 7
Line count: 2
```