Programming Test Results (With Test Cases)

Result Summary

Field	Value
Test ID	41308
Student ID	29135
Programs (with test cases)	3
Total Test Cases	11
Test Cases Passed	11
Fully Passed Programs	3
Partially Passed Programs	0
Failed Programs	0
Overall % (with test cases)	100.00%
Grade	Outstanding

Programs With Test Cases

#	Program Name	Total TC	Passed	Success Rate	Score /10	Submitted At	Attempts
1	VotingEligibility	3	3	100.0%	10	27/12/2025, 10:13:58 am	0
2	CreateDuplicateNumberException	4	4	100.0%	10	27/12/2025, 9:59:16 am	0
3	NoVowelsException	4	4	100.0%	10	27/12/2025, 9:46:03 am	0

Program Details (With Test Cases)

Program 1: VotingEligibility

Languages: java

Score (010): 10 / 10

Test Case Summary: Total: 3 Passed: 3 Failed: 0 Success: 100.0%

Attempts: 0

Submitted At: 27/12/2025, 10:13:58 am

Description:

Custom Exception Handling

Question:

Write a Java program that defines a custom exception called InvalidAgeException. In the main program, prompt the user to enter their age. If the age is less than 18, throw the custom exception with an appropriate message; otherwise, print "Eligible to vote".

Note :- take input from user using scanner class

Constraints: -

Sample Input: 18

Sample Output: Eligible to vote.

Explanation: -

Solution Code

```
import java.util.Scanner;
import java.util.InputMismatchException;
public class Test{
    void main(){
        Scanner sc = new Scanner(System.in);
        try(sc){
        int age = Integer.parseInt(sc.nextLine());
            if(age<18){
                throw new InvalidAgeException("Exception caught: Age "+age+" is not
valid. Must be at least 18 to vote.");
            } else IO.println("Eligible to vote.");
        }catch(InvalidAgeException e){
            IO.println(e.getMessage());
        }catch(NumberFormatException e){
            IO.println("Invalid input. Please enter a valid number.");
        }
    }
}
class InvalidAgeException extends java.lang.RuntimeException{
```

```
    public InvalidAgeException(){
        super();
    }
    public InvalidAgeException(String str){
        super(str);
    }
}
```

Program 2: CreateDuplicateNumberException

Languages:	Java	
Score (010):	10 / 10	
Test Case Summary:	Total: 4	Passed: 4
	Failed: 0	Success: 100.0%
Attempts:	0	
Submitted At:	27/12/2025, 9:59:16 am	

Description: Write any java program to read Array of integers from the user and throw an exception if the number is duplicate.
Note: Create DuplicateNumberException class as an unchecked exception.

Constraints: -

Sample Input: Enter number of elements: 5 Enter array elements: 1 2 3 4 5

Sample Output: No duplicate numbers found in the array.

Explanation: The program reads array elements using Scanner. A custom unchecked exception DuplicateNumberException is created by extending RuntimeException. The checkDuplicates() method compares every element with others using nested loops. If any duplicate value is found, the program throws the exception. Since it is an unchecked exception, it is not mandatory to handle using trycatch.

Solution Code

```
public class Test{
    void main(){
        int size = Integer.parseInt(IO.readln());
        int[] arr = new int[size];
        for(int i=0;i<size;i++){
            arr[i] = Integer.parseInt(IO.readln());
        }
        try{
            new Test().checkDuplicate(arr);
        }catch(DuplicateNumberException e){
            IO.println(e.getMessage());
        }
```

```
    }
    public void checkDuplicate(int[] arr){
        boolean isDuplicate = false;
        for(int i=0;i<arr.length;i++){
            for(int j=i+1;j<arr.length-1;j++){
                if(arr[i] == arr[j]){
                    isDuplicate = true;
                    throw new DuplicateNumberException("Exception in thread \"main\"
DuplicateNumberException: Duplicate number found: "+arr[i]);
                }
            }
        }
        if(!isDuplicate){
            IO.println("No duplicate numbers found.");
        }
    }
}
class DuplicateNumberException extends java.lang.RuntimeException{
    public DuplicateNumberException(){
        super();
    }
    public DuplicateNumberException(String s){
        super(s);
    }
}
```

Program 3: NoVowelsException

Languages:	Java	
Score (010):	10 / 10	
Test Case Summary:	Total: 4	Passed: 4
	Failed: 0	Success: 100.0%
Attempts:	0	
Submitted At:	27/12/2025, 9:46:03 am	
Description:	1) Write a java program to create a method that takes string as input and throw an exception if the string does not contains vowels. Note: Create NoVowelsException class as a checked exception.	
Constraints:	-	
Sample Input:	hello	

Sample Output: The string contains vowels.

Explanation: The program demonstrates custom checked exception handling in Java. It includes a custom exception class NoVowelsException that is thrown when a string does not contain any vowels. A method is created to check the input string for vowels. If no vowels are found, the exception is thrown; otherwise, it confirms that the string contains vowels. The main method uses a try-catch block to handle this exception and display appropriate messages. Key Points: NoVowelsException extends Exception to make it a checked exception. The method validates the string for vowels (a, e, i, o, u). Demonstrates the use of exception handling in Java (try-catch).

Solution Code

```
public class Test{
    public static void main(String[] args){
        String str = IO.readln();
        new Test().checkVowel(str);
    }
        public void checkVowel(String str){
            boolean isContainVowel=false;
            try{
                for(char ch : str.toCharArray()){
                    if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'||ch=='A'||ch=='E'||ch=
='I'||ch=='O'||ch=='U'){
                        System.out.println("The string contains vowels.");
                        isContainVowel = true;
                        break;
                 }
                }
                if(!isContainVowel) throw new NoVowelException("Exception caught: The
string does not contain any vowels!");
            }catch(NoVowelException e){
            System.out.println(e.getMessage());
            }
        }
}
class NoVowelException extends java.lang.Exception{
    public NoVowelException(){
        super();
    }
    public NoVowelException(String s){
        super(s);
    }
}
```