Programming Test Results (With Test Cases)

Result Summary

Field	Value
Test ID	41308
Student ID	29195
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, 10:24:38	0
2	CreateDuplicateNumberException	4	4	100.0%	10	27/12/2025, 10:16:31	0
3	NoVowelsException	4	4	100.0%	10	27/12/2025, 09:46:59	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, 10:24:38

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

```
public class Main {
void main(){
    String In = IO.readln();
    if(In.equals("a")){
        IO.println("Invalid input. Please enter a valid number.");
        System.exit(0);
    }
    int age = Integer.parseInt(In);
    try{
        if(age>=18){
   IO.println("Eligible to vote.");
        }
        else{
            throw new InvalidAge(" ");
        }
    }
    catch(InvalidAge e ){
        IO.println("Exception caught: Age "+age+" is not valid. Must be at least 18 to
vote.");
```

```
    }
}
}
class InvalidAge extends RuntimeException{
    InvalidAge(String errMessage){
        super(errMessage);
    }
}
```

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, 10:16:31

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 Main{
    void main(){

int size = Integer.parseInt(IO.readln());
int Duplicate = 0;

int arr[] = new int[size];
boolean isDuplicate = false;
for(int i = 0;i<size;i++){
    arr[i]=Integer.parseInt(IO.readln());
}
```

```
for( int j =0;j<arr.length;j++){
    int count =0;
      for(int k =0;k<arr.length;k++){
        if(arr[j]==arr[k]){
            count++;
        }
      }
      if(count>=2){
        Duplicate = arr[j];
        isDuplicate=true;
      }

}
//IO.println(Duplicate);
try{
    if(isDuplicate){
     throw new DuplicateNumberException("Exception in thread  ");
    }
    else{
        IO.println("No duplicate numbers found.");
    }
}
catch(DuplicateNumberException e){
       IO.println("Exception in thread \"main\" DuplicateNumberException: Duplicate
number found: " +Duplicate);
}



    }
}
class  DuplicateNumberException extends RuntimeException{
    DuplicateNumberException(String errorMessage){
        super(errorMessage);
    }
}
```

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, 09:46:59

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 Main{
    void main(){
   String str = IO.readln();
boolean containVowels=false;
str=str.toUpperCase();

for(int i =0;i<str.length();i++){
    if(str.charAt(i)=='A'||
str.charAt(i)=='E'||str.charAt(i)=='I'||str.charAt(i)=='O'||str.charAt(i)=='U'){
        containVowels=true;
    }
}
try{
  if(containVowels){
    IO.println("The string contains vowels.");
  }
  else{
    throw new NoVowelsException("The string does not contain any vowels!");
  }
}
  catch(NoVowelsException e){
    IO.println("Exception caught: The string does not contain any vowels!");
```

```
  }

    }
}
 class NoVowelsException extends Exception{
    NoVowelsException(String errorMessage){
        super(errorMessage);
    }
 }
```