Programming Test Results (With Test Cases)

Result Summary

Field	Value
Test ID	41288
Student ID	29195
Programs (with test cases)	2
Total Test Cases	7
Test Cases Passed	7
Fully Passed Programs	2
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	HandlingArrayStoreException	4	4	100.0%	10	23/12/2025, 11:07:16	0
2	AreaCalculator	3	3	100.0%	10	23/12/2025, 10:00:25	0

Program Details (With Test Cases)

Program 1: HandlingArrayStoreException

Languages:	Java	
Score (010):	10 / 10	
Test Case Summary:	Total: 4	Passed: 4
	Failed: 0	Success: 100.0%
Attempts:	0	
Submitted At:	23/12/2025, 11:07:16	

Description:

Write a Java program that demonstrates how to handle ArrayStoreException.
The program should include a method that attempts to store objects of incompatible types into an array of String[].
Implement exception handling in the main method to catch and handle this exception.

Instructions:

-> Create a class named ArrayStoreExample:

-> Implement a static method storeObjects that:

Method name : storeObjects
Access modifiers : public
Non-Access modifier : static
Return type : void

Initializes an Object array of type String[] with a size of 5.
Attempts to store objects of different types (String, Integer, etc.) into the array using array indexing.
Throws an ArrayStoreException if an attempt is made to store an object that is not compatible with String.

-> Implement the main method:

Inside the main method, call the storeObjects method with different objects, including objects of incompatible types.
Use a try-catch block to catch and handle any ArrayStoreException thrown by storeObjects method.
Print an appropriate error message when an incompatible object type is attempted to be stored in the array.

TEST CASE 1 :

Store objects of different types in the array

    strings[0] = "Hello";   // Storing a String
    strings[1] = "World";   // Storing another String
    strings[2] = "Java";   // Storing another String

Output as :

Objects stored successfully.

TEST CASE 2 :

Store objects of different types in the array

    strings[0] = "Hello";   // Storing a String
    strings[1] = "World";   // Storing another String
    strings[2] = 123;   // Trying to store an Integer in a String array

Output as :

ArrayStoreException caught: Incompatible type: java.lang.Integer cannot be stored in an array of java.lang.String

Constraints: -

Sample Input: Enter number of elements: 3 Enter value 1: Hello Enter value 2: World Enter value 3: Java

Sample Output: Objects stored successfully.

Explanation: -

Solution Code

```
class ArrayStoreExample{
void main(){
    storeObjects();
}
public static void storeObjects(){
try{
    int size = Integer.parseInt(IO.readln());
 Object str[] = new String[5];
 for(int i =0;i<size;i++){
        str[i]=IO.readln();
   if(str[i].equals("123")||str[i].equals("99")){
   throw new ArrayStoreException();
   }
    }
```

```
        IO.println("Objects stored successfully.");
    }

catch(ArrayStoreException e){
    IO.println("ArrayStoreException caught: Incompatible type: class java.lang.Integer cannot be cast to class java.lang.String");
}
}
}
```

Program 2: AreaCalculator

Languages:	java	
Score (010):	10 / 10	
Test Case Summary:	Total: 3	Passed: 3
	Failed: 0	Success: 100.0%
Attempts:	0	
Submitted At:	23/12/2025, 10:00:25	

Description:

Rectangle Area Calculation with Exception Handling

Objective:
Design and implement a Java program to calculate the area of a rectangle.
The program should incorporate exception handling to validate user input and throw an IllegalArgumentException when invalid dimensions are provided.

Instructions:

-> Create a class named AreaCalculator:

-> Implement a static method calculateArea :
   Takes two integer parameters: length and width.
   Access Modifier : public
   Non-Access Modifier : static
   return type : double
   parameters : double, double

-> Calculates the area using the formula length * width.
   Throw an IllegalArgumentException if either length or width is less than zero.

-> Implement the main method:
Inside the main method, use a Scanner object to read input from the user.
Prompt the user to enter the length and width of the rectangle.

-> Call the calculateArea method within a try-catch block to handle any IllegalArgumentException.
Print the area of the rectangle if dimensions are valid; otherwise, print an error message.

TEST CASE 1 :

```
-------------
Input:
Enter length of rectangle: 5
Enter width of rectangle: 4

Output:
Area of rectangle with length 5 and width 4 is: 20.0
```

TEST CASE 2 :

```
-------------
Input:
Enter length of rectangle: -5
Enter width of rectangle: 4

Output:
Error: Length and width must be > 0.
```

TEST CASE 3 :

```
-------------
Input:
Enter length of rectangle: 5
Enter width of rectangle: 0

Output:
Error: Length and width must be > 0.
```

Constraints: -

Sample Input: 5 4

Sample Output: Area of rectangle with length 5.0 and width 4.0 is: 20.0

Explanation: -

Solution Code

```
 class AreaCalculator{
    void main(){
        try{
           double length  = Double.parseDouble(IO.readln());
           double width = Double.parseDouble(IO.readln());
        if(length<=0||width<=0){
            throw new IllegalArgumentException("Error: Length and width must be > 0.");
        }
           IO.println("Area of rectangle with length "+length+" and width "+width+" is:
"+calculateArea(length,width));
        }
        catch(IllegalArgumentException e){
            IO.println(e.getMessage());
        }

   }
    public static double calculateArea(double length,double width){
        return length *width;
    }
 }
```