Contents

5.4 Exercises for lab

Students are provided the Checklist and they must identify and explore each type of general errors that may arise during inspection session. Inspection Checklist of Errors: Data Reference

Computational error Checklist

Checklist Item	Error Found
1. Computations on nonarithmetic variables	Mixing arithmetic with non-arithmetic
2. Mixed-mode computations	Mixing integer with double
3. Computations on variables of different lengths	Mixing short with int
4. Target size less than size of assigned value	Target size is less than assigned value
5. Intermediate result overflow or underflow	Overflow
6. Division by zero	Division by zero

Version 1: Improved Error Handling

- Version 1 directly handles division by zero using try-catch

Test Cases for Version 1: Improved Error Handling

Test ID	Test Case Description	Input Data	Actual	Expected	Verdict
1	Computations on nonarithmetic variables	"Hello"	10	10	Passed
2	Mixed-mode computations		15.5	15.5	Passed
3	Computations on variables of different lengths		1100	1100	Passed
4	Target size less than size of assigned value		10	10	Passed
5	Intermediate result overflow or underflow		2147483648	2147483648	Passed
6	Division by zero		Error: Division by zero	Error: Division by zero	Passed

Code:

Main.java

```java
public class ComputationChecklistV1 {
    public static void main(String[] args) {
        // 1. Computations on nonarithmetic variables
        String nonArithmeticVariable = "Hello";
        int result1 = 5 + nonArithmeticVariable.length();

        // 2. Mixed-mode computations
        double integerVariable = 10;
        double doubleVariable = 5.5;
        double result2 = integerVariable + doubleVariable;

        // 3. Computations on variables of different lengths
        int shortVariable = 100;
        int intVariable = 1000;
        int result3 = shortVariable + intVariable;

        // 4. Target size less than size of assigned value
        int[] array = new int[1];
        array[0] = 10;

        // 5. Intermediate result overflow or underflow
        long maxValue = (long) Integer.MAX_VALUE + 1; // Use long to avoid overflow
        long result4 = maxValue;

        // 6. Division by zero
        int denominator = 0;
        int result5 = 0;
        try {
            result5 = 10 / denominator; // Try to perform division
        } catch (ArithmeticException e) {
            System.out.println("Error: Division by zero"); // Handle division by zero error
        }
        // Print results
        System.out.println("Result 1: " + result1);
        System.out.println("Result 2: " + result2);
        System.out.println("Result 3: " + result3);
        System.out.println("Result 4: " + result4);
        System.out.println("Result 5: " + result5);
    }
}
```

Version 2: Using Data Validation

- Version 2 uses a separate method to perform division safely by validating the denominator.

Test Cases for Version 2: Using Data Validation

Test ID	Test Case Description	Input Data	Actual	Expected	Verdict
1	Computations on nonarithmetic variables	"Hello"	10	10	Passed
2	Mixed-mode computations		15.5	15.5	Passed
3	Computations on variables of different lengths		1100	1100	Passed
4	Target size less than size of assigned value		10	10	Passed
5	Intermediate result overflow or underflow		2147483648	2147483648	Passed
6	Division by zero		Error: Division by zero	Error: Division by zero	Passed

Code:

Main.java

```java
public class ComputationChecklistV2 {
    public static void main(String[] args) {
        // 1. Computations on nonarithmetic variables
        String nonArithmeticVariable = "Hello";
        int result1 = 5 + nonArithmeticVariable.length(); // Compute the length of the string

        // 2. Mixed-mode computations
        double integerVariable = 10; // Change to double to match doubleVariable
        double doubleVariable = 5.5;
        double result2 = integerVariable + doubleVariable;

        // 3. Computations on variables of different lengths
        int shortVariable = 100;
        int intVariable = 1000;
        int result3 = shortVariable + intVariable;

        // 4. Target size less than size of assigned value
        int[] array = new int[1];
        array[0] = 10;

        // 5. Intermediate result overflow or underflow
        long maxValue = (long) Integer.MAX_VALUE + 1; // Use long to avoid overflow
        long result4 = maxValue;

        // 6. Division by zero
        int denominator = 0;
        int result5 = divideSafely(10, denominator); // Use a method to handle division safely

        // Print results
        System.out.println("Result 1: " + result1);
        System.out.println("Result 2: " + result2);
        System.out.println("Result 3: " + result3);
        System.out.println("Result 4: " + result4);
        System.out.println("Result 5: " + result5);
    }

    // Method to perform division safely
    private static int divideSafely(int numerator, int denominator) {
        if (denominator == 0) {
            System.out.println("Error: Division by zero");
            return 0; // Return 0 if division by zero
        }
        return numerator / denominator; // Perform division if denominator is not zero
    }
}
```