System Requirements Specification Index

For

Arithmetic Operators

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

USE CASE DESCRIPTION

System Requirements Specification

1 Project Abstract

Create a Java program that demonstrates the use of arithmetic operators: +, -, *, /, and %. You will use two variables of different types (int and double), perform basic arithmetic operations on them, and print the results.

2 Assessment Tasks

1. **Declare 2 variables:**
   - A variable named `a` of `int` datatype, initialized with the value `10`.
   - A variable named `b` of `double` datatype, initialized with the value `20.5`.
2. **Perform Arithmetic Operations:**

   Perform the following arithmetic operations on the declared variables:

   - **Addition**:
     1) Add `a` and `b` after casting `b` to an integer.
     2) Store the result in a variable named `sum` of `int` datatype.
   - **Subtraction**:
     1) Subtract `a` from `b`.
     2) Store the result in a variable named `difference` of `double` datatype.
   - **Multiplication**:
     1) Multiply `a` and `b` after casting `b` to an integer.
     2) Store the result in a variable named `product` of `int` datatype.
   - **Division**:
     1) Divide `b` by `a`.
     2) Store the result in a variable named `quotient` of `double` datatype.
   - **Modulus (remainder)**:
     1) Find the remainder of dividing `a` by `3`.
     2) Store the result in a variable named `remainder` of `int` datatype.
3. **Print the Results:**
   - Print the results of the arithmetic operations i.e, `sum`, `difference`, `product`, `quotient` and `remainder` with appropriate labels in separate lines as shown in the expected output.

4. **Expected Output:**

Sum: 30

Difference: 10.5

Product: 200

Quotient: 2.05

Remainder: 1

3 Template Code Structure

3.1 Package: com.yaksha.assignment.ArithmeticOperatorsAssignment

Resources

Class/Interface	Description	Status
ArithmeticOperatorsAssignment (class)	• Main class containing the logic to perform arithmetic operations (+, -, *, /, %) on variables.	Need to be implemented.

4 Execution Steps to Follow

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) Terminal New Terminal.**
3. **This editor Auto Saves the code.**
4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**

6. **To run your project use command:**
   mvn compile exec:java
   -Dexec.mainClass="com.yaksha.assignment.ArithmeticOperatorsAssignment"
7. **To test your project test cases, use the command**
   mvn test

8. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**