Project 7a

Create a class that has methods to perform each of the arithmetic operations on two numbers.

-Include in the class a main method as well as methods that perform the arithmetic operations.

-In main, prompt the user for the input of two numbers. After storing the values call each of the four methods that perform simple arithmetic operations.

-Each method should accept two numbers as formal parameters.

-The addition method should add the two numbers together and display the sum with a descriptive statement. The return type will be void.

-The subtraction method should subtract the smaller from the larger and display the difference. (Math.min() and Math.max() might help)

-The multiplication method should multiply the two numbers together and return the product to be displayed in main.

-The division method should check for division by zero and if the second number is zero return a string error message otherwise the method should return a String that reports the quotient and modulus along with descriptive labels. The string, error or report, will be displayed in main.

-Make sure and validate all user inputs as well as providing a testing grid to validate program output.

|  |  |  |  |
| --- | --- | --- | --- |
| Input (num1, num2) | Expected Results (add, subtract, multiply, quotient, modulus) | Actual Results (add, subtract, multiply, quotient, modulus) | Reason if Different |
| 1,1 | 2.0, 0.0, 1.0, 1, 0 | 2.0, 0.0, 1.0, 1, 0 | None |
| 5,3 | 8.0, 2.0, 15.0, 1, 2 | 8.0, 2.0, 15.0, 1, 2 | None |
| One,2 | Format Error | Format Error | None |

Evaluation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Aspect | Objectives substantially met  90 – 100% | Meets Minimal Requirements  80-89% | Needs Improvement  79 – 79% | Failure to Meet Requirements  0 - 69% |
| Good Programming Practices  10% | Effective use of white space.  Clear and appropriate documentation.  Implements all needed error handling.  Data types and identifiers meet all expectations. | Use of white space and documentation with minor defect. Implementation of error handling general but lacking in minor aspects.  Selection of data types and / or creation of identifiers not consistent. | Generally meets expectations for good programming practices. | In the main does not meet expectations for good programming practices. |
| Iterative structures  10% | Iterative structure appropriate to the processing needed including validation of all inputs | Iterative structure appropriate to the processing needed but validates only some inputs | Iteration for processing appropriate but input validation not performed | Iteration not used |
| Modularization  40% | Methods properly declared with an appropriate parameter list and provide appropriate return.  Method calls pass parameters correctly | Minor defects in definition of methods or the passing of parameters or return invalid, | Needs improvement in method declaration, method calls, return statements or parameter passing | Fails to properly modularize the algorithm or parameter lists not according to specs or specified return not made |
| Accuracy of Output  40% | All output correct and supported with appropriate testing. | With minor exceptions the output is correct and testing may be missing some needed test cases. | Only part of the output is correct as a result of inadequate test cases. | Only some of the output is correct as a result of missing or misused test cases. |