Calculator

Create a class (or user-defined type) called **Calculator** that might be used to perform a simple arithmetic operation (+, -, ×, /). Define the type **Calculator** as follows:

a. Declare private member variables for an array to hold every value entered your calculator, a variable that tracks the total numbers entered, and a variable for an answer. The maximum number of the values that can be entered into your calculator is 4.
b. Declare/define a public default constructor that initializes the private member variables of each object with 0 when an object of the class **Calculator** is created.
c. Declare/define a public void member function for addition without any parameter.
d. Declare/define a public void member function for subtraction without any parameter.
e. Declare/define a public void member function for multiplication without any parameter.
f. Declare/define a public void member function for division without any parameter.
g. Declare/define a public void member function that takes in a number and saves the number in the internal array. The member variable storing the total number entered should be updated whenever a number is saved (or entered). If the array is full, the function should print an error message as shown in the test cases and NOT save the number to the internal array.
h. Declare/define a public void member function that clears all values from the calculator's memory.

Document these member functions with appropriate comments. Embed your class **Calculator** in your program (main) that requests data (see the sample output) and tests the member functions of the Calculator class.

To use the calculator, at least two numbers must be available in memory before an operation can be performed. At most four numbers can be entered by the user. When an answer is found, it becomes the only value in the calculator (see the sample output). In your program (main), create a switch statement in a loop which allows you to select options on your calculator. Have an option to quit.

The program should present a menu with seven options (see the sample output). It should display an error message like "Invalid input. Try again!" if what is entered is not one of the options (see the sample output) and display the menu again.

The following sample output is what your program should look like. Note that red characters/numbers are examples of the user inputs.

Sample Output

The following test cases do not cover all possible scenarios but should indicate if your code is on the right track. **To guarantee full credit, your program's output should exactly match the output**

below.

Test Cases:

CALCULATOR
What would you like to do?

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: b

Invalid input. Try again!

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: a

Enter at least 2 numbers for calculation!

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: -16

e) Enter number
a) Add
s) Subtract

m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: 4.21

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: 17

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: a

-16 + 4.21 + 17 = 5.21

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: 2.2

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: 1.5

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: s

5.21 - 2.2 - 1.5 = 1.51

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: 10

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: 5

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: d

1.51 / 10 / 5 = 0.0302

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: 20

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: 1000

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator

```
q) Quit

Choice: m

0.0302 * 20 * 1000 = 604

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: -1

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: m

604 * -1 = -604

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: c

All values are cleared!

e) Enter number
a) Add
s) Subtract
```

m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: a

Enter at least 2 numbers for calculation!

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: 10

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: 20

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: 30

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: 40

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: e

Enter a number into calculator: 50

Failed to enter the current number!
No room left for a new number!

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator
q) Quit

Choice: s

10 - 20 - 30 - 40 = -80

e) Enter number
a) Add
s) Subtract
m) Multiply
d) Divide
c) Clear calculator

q) Quit

Choice: q

Quitting... Goodbye!

Your grade will be subject to the following condition(s):

- Submission:
  Late submission will NOT be accepted, so be sure to start early and submit early.

  Submit your code on Canvas. You just need to **upload** a .cpp file. Please name your .cpp file using your **2 initials** followed by **M8HWA** (for example, **kc_M8HWA.cpp**)

  PLEASE doublecheck your file has actually been uploaded.

Your grade will be calculated based on the following (total 100 points):

- Initial comments (short description, inputs, outputs, your name): **10 pt**
- Compilation: **30 pts**
  Your source code **MUST** be complied successfully without error. There is no partial credit available here, either your code compiles or it doesn't.
- Execution: **40 pts**
  Your program must include and test the user-defined class (**Calculator**) that contain all the instructed members (member variables, constructors, member functions) (20 points). It will be tested against 8 cases, each worth 2.5 points. You can earn partial credit here if your code doesn't work for every single case. If it does work for every case, you will get the full 20 points.
- Style: **20 pts**
  Your code will also be graded on its style. This includes things like using meaningful variable names, useful comments, proper indentation and spacing, the proper use of functions, and the proper use of user-defined class type. All of these things make your code easy to read and maintain. Partial credit will be available here. At a minimum, your code should have a comment at the beginning with your name, date, and a high level but still descriptive overview of what the program does.