**Planning sheet**

**Task 1: Describe the problem**

*Write a menu-driven program which asks for 2 integers from the user, then presents a menu of options*

*(Add, Subtract, Multiply, Divide, Quit). The user chooses an option, the program performs the*

*operation then asks for another choice unless the choice was Quit.*

**Task 2: Identify the input information**

What information will the user have to enter? Copy and complete this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable name** | **Scope** | **Data type** | **Purpose of variable** |
| integer\_one | global | int | storing the first number to do maths with |
| integer\_two | global | int | storing the second number to do maths with |
| opt\_choice | local (??) | float | choosing an option in the menu (add is a, subtract is s, etc.) |

**Task 3: Identify other variable information**

What information will the program need to store? Also include constants and derived values as they will maximise the flexibility and robustness of your plan. Your chosen scope should match the way the variable will be used in your program.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable name** | **Scope** | **Data type** | **Purpose of variable and where it comes from.** |
| result | local but gets returned to print | int (or float for division) | there’ll be one in addition, subtraction, multiplication, and division. it stores the result of the equation performed on the two integers |

**Task 4: Identify the indexed data (list) structure**

What information will be stored in a list?

|  |  |  |  |
| --- | --- | --- | --- |
| **List name** | **Scope** | **Data type** | **Purpose** |
| menu options | local | string | holds all the options of the menu, possibly 2d? ([Addition, a], [Subtraction, s], etc.) |

**Task 5: Identify the output information**

What information will the program need to print out?

|  |  |  |
| --- | --- | --- |
| **Output** | **When it happens** | **Details** |
| the equation and the result e.g. ({} + {} = {}).format(first number, second number, result - call the function for it) | after the user has chosen a menu option and the calculation has been performed in the respective function | outputs three numbers - the numbers the user chose and the result of whatever calculation they chose |
| Farewell message | When the user presses q | Something like ‘thank you!’ or ‘goodbye’. |
| Please enter valid input | If the user enters something they’re not supposed to | Could vary depending on what they inputted and when |

**Task 6: Write the algorithm**

1. call force\_int function - input - ask for the first integer, store in variable
2. call force\_int function - input - ask for the second integer, store in variable
3. **start infinite while true loop**
4. call menu function
   * will output options from list - (Add - a, Subtract - s, etc.)
   * **start infinite while true loop**
   * if the input is in the list of correct options (a, s, m, d, q), **break loop** and return the option
   * else, print error message and let the loop repeat
5. if opt\_choice == a:
   * call addition function
     + parameters are the two integers
     + adds them together and then returns the sum
   * print out the sum
6. if opt\_choice == s:
   * call subtraction function
     + parameters are the two integers
     + subtracts the second one from the first and then returns the result
   * print out the result
7. if opt\_choice == m:
   * call multiplication function
     + parameters are the two integers
     + multiplies them together and returns the product
   * print out the product
8. if opt\_choice == d:
   * call division function
     + parameters are the two integers
     + divides the first by the second and returns the result
   * print out the result
9. if opt\_choice == q:
   * print out thank you/farewell message
   * **break loop**

**Task 7: Testing Plan**

|  |  |
| --- | --- |
| **TEST** | **EXPECTED RESULT** |
| **Integer selection** | |
| **expected positive ints: 3, 10, 78, etc.** | **should take in the integer and use it without problems** |
| **expected negative ints: -5, -12, -62, etc.** | **should take in the integer and use it without problems** |
| **unexpected floats: 3.14, 0.25, etc.** | **should come up with an error message and ask them to enter a whole number** |
| **unexpected strings: hello, three, a, etc.** | **should come up with an error message and ask them to enter a whole number** |
| **(nothing, the user pressed enter)** | **should come up with an error message and ask them to enter a whole number** |
| **Menu selection** | |
| **expected choice: a** | **should start the addition function and return the equation with the answer** |
| **expected choice: s** | **should start the subtraction function and return the equation with the answer** |
| **expected choice: m** | **should start the multiplication function and return the equation with the answer** |
| **expected choice: d** | **should start the division function and return the equation with the answer (rounded to 2dp)** |
| **expected choice: q** | **should break the loop and end the program** |
| **unexpected choice string: h, seven, addition, etc.** | **should come up with an error message and ask them to enter a valid choice** |
| **unexpected choice number: 32, 6.7, etc.** | **should come up with an error message and ask them to enter a valid choice** |
| **(nothing, the user pressed enter)** | **should come up with an error message and ask them to enter a valid choice** |
| **Calculations** | |
| **Addition positive ints: 3 and 7, 40 and 32, etc.** | **3 + 7 = 10, and 40 + 32 = 72** |
| **Addition negative ints: -4 and 6, -5 and -8, etc.** | **-4 + 6 = 2, and -5 + -8 = -13** |
| **Subtraction positive ints, int1 > int2: 5 and 2, etc.** | **5 - 2 = 3** |
| **Subtraction positive ints, int1 < int2: 2 and 5, etc.** | **2 - 5 = -3** |
| **Subtraction negative ints: 4 and - 6, -3 and 2, etc.** | **4 - -6 = 10, and -3 - 2 = -5** |
| **Multiplication positive ints: 5 and 4, 2 and 8, etc.** | **5 \* 4 = 20, and 2 \* 8 = 16** |
| **Multiplication negative ints: -2 and 6, -7 and -4, etc.** | **-2 \* 6 = -12, and -7 \* -4 = 28** |
| **Division positive ints, int1 > int2: 10 and 2, 5 and 4, etc.** | **10 / 2 = 5.00, and 5 / 4 = 1.25** |
| **Division positive ints, int1 < int2: 3 and 6, 2 and 10, etc.** | **3 / 6 = 0.5, and 2 / 10 = 0.2** |
| **Division negative ints: -20 and 4, -21 and -7, etc.** | **-20 / 4 = -5, and -21 / -7 = 3** |