Working with Multiples

So far, our programs have been working with single values. In this module you will see how we can use the power of the computer to work with multiples by making use of arrays.

Learning Goals

To complete this task, you need to demonstrate that you can do the following:

- Declare and use arrays within your code in variables, fields, and parameters.
- Write code that interacts with the array, and with its elements.
- Iterate over the elements in an array to interact with each as an individual value.
- Use hand execution to illustrate how code works with an array.

Focus

- **Programming concepts:** Focus on how an array lets you store and work with multiple values, and how this works with loops.
- **Programming process:** Focus on separating the processing of individual elements and processing all elements in the array and see how these come together using loops.
- **Syntax**: Focus on memorising the syntax for declaring arrays and accessing individual elements. Also remember the limitations that C introduces, and how we can work around these using structs.
- **Professional characteristics:** Focus on your persistence in this task. Understanding arrays may take some back and forth, rereading and testing out your understanding until it makes sense.

Your Task

For this task you will need to submit the following:

- A PDF document containing:
  - Learning Journey and Resources
  - Screenshots of your statistics program running.
  - At least five (5) photos of your hand execution on paper
  - Summaries and Reflections
- Source code for:
  - The Statistics Calculator
  - Your chosen program from the Test Your Knowledge activities

1. Complete Learning Activities

Work through these steps to develop and demonstrate your understanding. Aim to demonstrate, to yourself and others, that you have achieved the learning goals.

1. Everything you need is in Chapter 5 Working with Multiples from Part 2 of the Programmer's Field Guide.
2. Review details on the arrays, accessing array values using loops, and arrays within structs.
3. Build at least three small programs to test out how arrays work
   a. Create a small array, access each value directly using literal values for the indexes.
   b. Update your program, or create a new one, to test out using variables to access values one at a time inside a loop.
   c. Create a small program where you use a struct to contain your array – with the struct storing an integer for the number of elements that have values, and an array with up to 10 values. Test

its use by creating functions and procedures that work on your struct.

4. Build the Statistics Calculator program. Make sure to build this iteratively – showing each iteration in your submission. **Capture** notes on your learning as you progress, indicating if and where you need to review the existing solutions.
5. Hand execute the following with your version of these functions to verify they work as intended. To simplify this, assume `MAX_NUMBERS` is set to 4, so the array in the number data has 4 elements at most.
    a. Maximum of [-7, -1, -10]
    b. Add Data 8 to [0, 5, -2]
    c. Remove Data at index 1 from [1, 2, 3]
    d. Remove Data at index 2 from [1, 2, 3]
    e. Remove Data at index -1 from [1, 2, 3]
6. Complete one of the Test Your Knowledge activities.
7. Prepare your summary, making sure to cover all learning goals and related concepts. Remember that this is a personal summary that demonstrates your understanding of the concepts.
8. Prepare your reflections by responding to the following:
    - How do you know you have achieved the learning goals?
    - What is the most important thing you learned from this and why?
9. Capture your learning journey and collate your evidence of study and practice.

2. Upload Your Submission & Engage with Feedback

Mark the task as **Ready for Feedback** and upload the required files. Make sure to keep copies of these in case you need to resubmit. Then engage with the feedback you receive and get the task Complete!

If you are asked to resubmit, make sure your subsequent submission includes a comment that describes how you have addressed the feedback you received. This needs to demonstrate how you have addressed all the aspects indicated by your tutor in their feedback on your learning. If you don't understand the feedback, ask for clarification. If it is too generic, ask specific questions, only you know what feedback you need, take charge of it.