

Sequence and Data

Our first programs will explore the creation of programs that involve a sequence of instructions that can manipulate data within the computer.

Learning Goals

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

- Create programs that involve sequence and data.
 - Indicate your code is using functionality from a library.
 - Code a sequence of actions that achieve a desired result.
 - Declare and use variables to store changing values, selecting appropriate data types to store the desired information.
 - Use expressions to calculate values using literals and variables.
 - Call methods, passing in required arguments.

Focus

As you work through this task, focus on the following aspects of the unit:

- **Programming concepts:** Focus on how the concepts are all connected, and each play a role in your program.
- **Programming process:** Programs requires an iterative process, where you develop your solution a little at a time. Focus on using this process to help you build up your programs a little at a time.
- **Coding:** Focus on memorising the syntax for each of the different instructions that you learn. You want to be able to recall these without having to look up your details.
- **Professional Characteristics:** Keep focusing on attention to details and building a strong understanding of the programming concepts. These characteristics will help you greatly as you progress.

Your Task

For this task you will need to submit the following:

- A PDF document containing:
 - Summaries and reflections.
 - At least three (3) screenshots :
 - One showing each of the following programs running on your computer
 - Learning Journey and Resources
- Source code for:
 - Change Calculator
 - Airspeed Velocity or Water Bottle Visualiser
 - The 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 0 Introducing Instructions](#) and [Chapter 1 Sequence and Data](#) from Part 1 of the [Programmer's Field Guide](#).

Tip

Remember to actively read these chapters. Build and run sample programs to test your understanding. Create notes, cheat sheets, diagrams, and/or concept maps to help you remember, connect and understand these concepts.

2. Build the [Change Calculator](#) and either the [Airspeed Velocity](#) or [Water Bottle visualiser](#). **Capture** notes on your learning as you progress, indicating if and where you need to review the existing solutions.

Tip

Remember that documenting your learning journey is an important part of the task. Capture the issues and successes you have as you go, watching out for *light bulb* moments where things suddenly all come together.

3. Complete one of the [Test Your Knowledge](#) activities from the Sequence and Data chapter of the Programmers Field Guide.
4. 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.
5. 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?
 - How does the content or skills learned here relate to things you already know?
6. Capture your learning journey and collate your evidence of study and practice.

Tip

Make sure to include any other programs and code snippets you created to help you learn this topic in your learning journey. These can be the small programs you created as you read through the chapter, even if they are just a few lines of code testing out an idea. Learning to actively read and test out programming ideas will be a useful skill that you can use when learning any new programming tool or technique.

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!