

# Project 1 – Structured Programming

Now that you understand control flow, the key mechanics of structured programming, it is time to test out using these to build a small program.

## Learning Goals

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

- Integrate everything you have learnt about sequence and data and control flow to build a small program.

## Focus

- **Programming concepts:** Focus on how you apply the concepts from sequence and data, and control flow to build a small program.
- **Programming Process:** Focus on using an iterative process to build your program.
- **Coding:** Focus on applying coding conventions associated with indentation and naming to produce professional quality code.
- **Professional characteristics:** Focus on showing you can use these tasks to build and strengthen your programming competencies, and that you are responsive to the feedback you have received.

## Your Task

For this task you will need to submit the following:

- A PDF document containing:
  - Your design notes and plan for your code.
  - Screenshots of your program running on your computer
  - Reflections and documentation of your learning journey
- Source code for your chosen program from the [Test Your Knowledge](#) activities.

### 1. Complete Learning Activities

1. Everything you need is in [Chapter 4 Projects](#) from Part 1 of the [Programmer's Field Guide](#).
2. Complete one of the [Test Your Knowledge](#) activities from the Structured Programming Project chapter of the Programmers Field Guide. **Capture** your plans for the design of your solution, your final code, and a screenshot of this working.

#### Tip

Your plans can include flow charts, written notes, UML diagrams, and/or pseudocode. Make sure to think through your solution before you start coding. This is an important skill to develop, so these notes should help demonstrate how you are building this capability. You want to plan to the point where you can see how it will all come together, and already envision what methods and data you may require.

Key is that your program should clearly demonstrate your understanding each concept.

3. Prepare your reflections by responding to the following:
  - How confident are you that you can now build small programs, and what gives you this confidence?
  - What are the areas you need to focus on next?
4. 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.

At the Credit level, you should demonstrate you can be proactive. To get the task Complete, you need to engage with the feedback. Identify one thing you would like to get feedback on and communicate this to your tutor via the feedback dialogue at the moment of submission.

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.

### Tip: Evaluate yourself

When you submit your work, you can ask your tutor for specific feedback. Try identifying what you have done well and what you have doubts about or would like feedback and clarification on. The feedback process is meant to help you learn and understand your strengths and weaknesses, evaluate yourself and get comfortable at asking for advice.