

# Control Flow

Programs need to include more than a sequence of instructions and data. Control flow statements provide you with the tools you need to take your programs to the next level – making them interactive and responsive.

## Learning Goals

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

- Work with Boolean data, including using comparisons and logical operators to form expressions that test given conditions.
- Use computational thinking to design and build programs that include branching and looping statements to implement a range of dynamic behaviours.

## Focus

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

- **Programming concepts:** Focus on computational thinking and algorithm design, and how this is achieved with Boolean logic and structured programming statements with sequence, branches, and loops.
- **Programming Process:** Focus on organising your code in blocks, breaking your code and your thinking down into different parts that all come together using sequence, branches, and loops.
- **Coding:** Focus on memorising the control flow statements, and how indentation is used to communicate the structure of your code.
- **Professional Characteristics:** Make sure to persist with this task and develop a good understanding of these concepts and how to apply them. Work on your ability to compartmentalise your thinking.

**Tip**

This is a big and important topic! You will need to dedicate some time to this, and you need to make sure that you really understand these ideas as they are central to everything in programming. Ensure you make use of all the support available to help you learn to use these new tools.

## Your Task

For this task you will need to submit the following:

- A PDF document containing:
  - Summaries and reflections.
  - Photos of your two hand executions (2 photos)
  - Screenshots of the following programs running
  - Learning Journey and Resources
- Source code for:
  - The revised Change Calculator.
  - Fly Catch.
  - Your chosen program from the [Test Your Knowledge](#) activities

### 1. Complete Learning Activities

Aim to demonstrate, to yourself and others, that you have achieved the learning goals.

1. Everything you need is in [Chapter 2 Communicating Syntax](#) and [Chapter 3 Control Flow](#) from Part 1 of the [Programmers Field Guide](#).

**Tip: Build small test programs**

Remember to actively read Chapter 3. I suggest you build a small test program for each control flow mechanism. You can then test each one out on its own. These small programs should be an important feature in your learning journey.

**Tip: Do this with others!**

Designing and building small test programs is a great group activity. Work with a group in class, and you can help each other understand how to think and work with these new concepts.

2. Demonstrate how the following two small programming snippets run using the [Hand Execution](#) process described in the Programmers Field Guide. Do this on paper with a pencil. **Capture** a photo of your workings.

---

```
using static System.Console;
using static System.Convert;

int target;

Write("Count to: ");
target = ToInt32(ReadLine()); // User enters 3

int i = 0;

while (i < target)
{
    WriteLine(i);
    i++;
}
```

---

```
using static System.Console;
using static System.Convert;

int i;

for (i = 0; i < 11 ; i++;)
{
    if ( i > 0 && i % 3 == 0 )
    {
        i++;
        continue;
    }

    WriteLine(i);
}
```

3. Build the [revised Change Calculator](#) and the [Fly Catch](#) program (at least up to the end of the [Move Spider](#) iteration). **Capture** notes on your learning as you progress, indicating if and where you need to review the existing solutions.

4. Complete one of the [Test Your Knowledge](#) activities from the Control Flow chapter of the Programmers Field Guide.

**Tip: A bit stuck? Be sure to reach out!**

Your tutors are programming experts and are there to help. Be sure to reach out to them during class or ask a question on the SIT102 MS Teams channel or the Discussion forums on the CloudDeakin site.

5. 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.
6. 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?
7. 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.