

Software Engineer

In collaboration with



Tech Council
of Australia

Overview

A Software Engineer, sometimes called a Software Developer or Computer Programmer, is someone who creates computer programs. They use code to design, build, test, troubleshoot and maintain computer software.

There are three main types of software, and Software Engineers deal with all of these (but often specialise in one). There's system software, which is the software that allows computers to run, like Windows, macOS, Android and iOS operating systems.

Then, there's the software that keeps your computer running and optimises programs. This is called utility software and includes tools like system clean-ups, file compression, and ant-virus software.

And finally, there's application software. This is the form of software people interact with the most, like websites, apps, word processors and spreadsheets.

At CommBank, a Software Engineer could be working on a range of different things, from their website or the CommBank App, to their internal systems that allow the bank to run behind the scenes – delivering a safe, secure and smooth digital experience for customers.

A Day in the Life

Depending on the business you work for and the type of software you specialise in, your average day as a Software Engineer will vary a fair bit. However, some typical duties may include:

Writing code

One of the main duties of a Software Engineer is writing code to create new software or add new features to existing programs.

Testing and debugging code

Once code has been written, engineers will then test the software to make sure it runs smoothly. If it doesn't, they need to find the cause and solve the problem. Software Engineers will also debug code to resolve technical issues for staff members, clients or customers.

Documenting software code and processes

Whenever a developer adds new commands or code to the software they're working on, they also note down explanations and instructions about how the software works – kind of like a user manual.

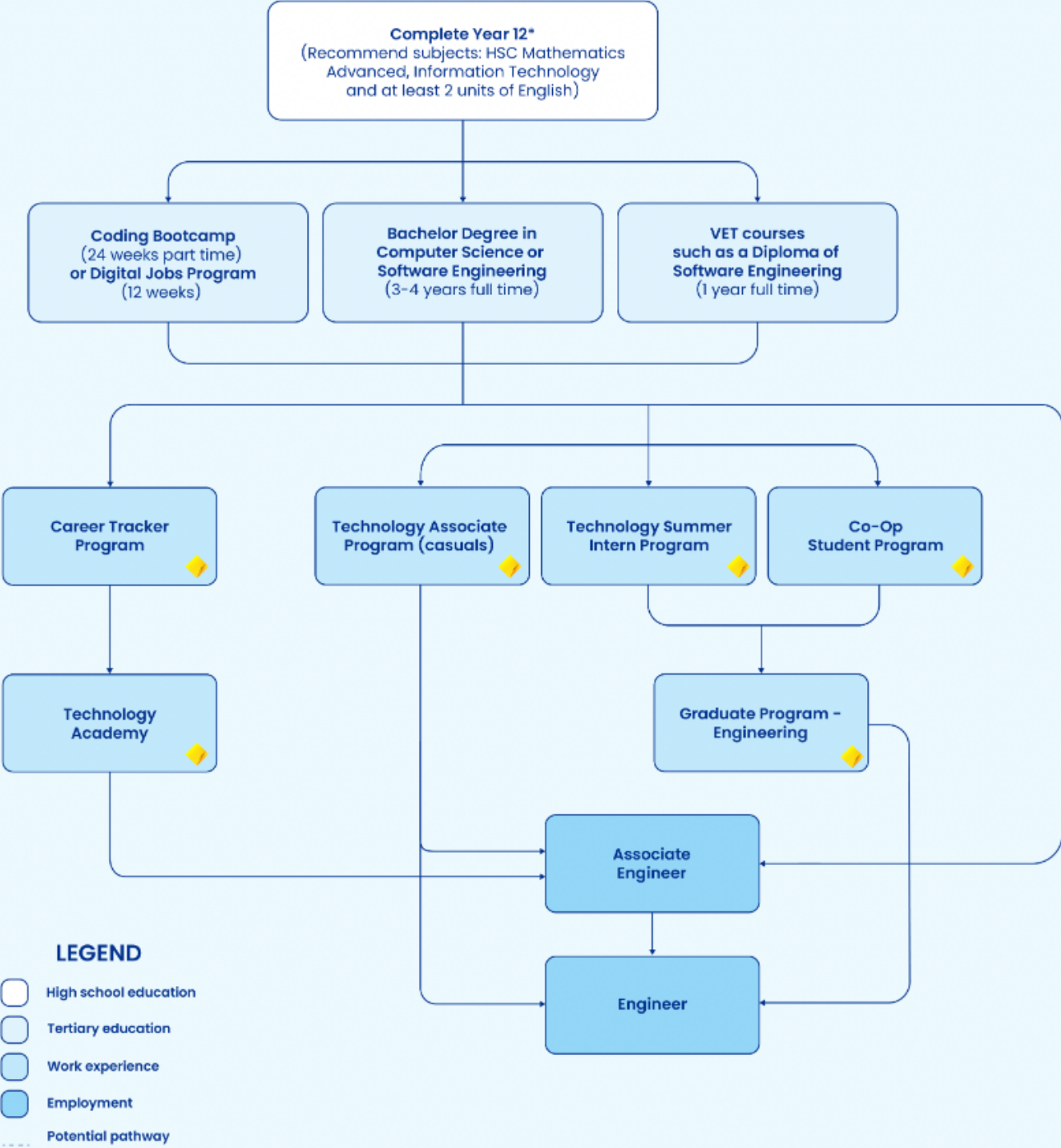
Working with other engineers to collaborate on projects

Teamwork and collaborating are a massive part of software development. If one member of a team specialises in one area and another team member is an expert in another, they'll work together to build the software as a whole. For example, one person might specialise in the ease of use of a program for customers, while another developer might be better at designing the aesthetics of software.

Learning new technologies and programming languages.

Right now, there are more than 700 coding languages used to develop software, and new coding languages are being developed all the time. Technology is no different, just look at the way AI is changing the world. This is why it's important for engineers to stay up to date with emerging codes which could take off – and being ready when they do.

Pathways



Activities

Activity 1: Understanding Coding Commands / Debugging Code

Introduction

A coding command is an instruction you give to a computer to make it do something. It's like that icebreaker game they do on school camps where your friend is blindfolded, and you have to give them directions to complete an obstacle course. But instead of using your voice, you write in code.

Creating commands is a foundation of software engineering. It's how websites, apps and other programs know how to run. Commands are like the brain of software – in the same way your brain tells your body what to do, commands do the same for programs.

Activity

Now it's your chance to learn the basics of creating your own commands. Scratch is a website home to games, designs and animations created by young coders, with dedicated lessons for beginners. For now, you're going to start by learning how to use commands to create simple instructions. Follow the lesson and read the instructions carefully. If you don't get it right on your first try, don't stress, there's a lot of trial and error involved in writing commands, which will mean testing and debugging (editing the commands to fix any errors). If you're struggling to understand the activity, you can take some time to read more about commands before you begin.

Activity Link <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

Conclusion

Writing commands is a step-by-step process. You have to think about it as if it were a story. One thing happens after another to reach the desired function. Writing commands and debugging code are a major aspect of work as a software developer. It's a bit like digital detective work, investigating where the errors in the code are and solving them so it runs smoothly again.

Activity 2: Create A Pong Game Using Commands

Introduction

All software is created using commands in the form of code. The commands serve a specific purpose to create functioning software. In the same way, you're about to use commands to create a fully functioning game.

Activity

For this activity, you're going to follow a Scratch tutorial to create a pong game. The commands will be more complex than in the first activity, but you will have instructions as you go. If you have any ideas to make the game more enjoyable as you go, feel free to play around and add them in!

Activity Link <https://scratch.mit.edu/projects/editor/?tutorial=pong>

Conclusion

Commands need to be perfect for software to run properly. If one piece of code is wrong, the software won't run the way it's supposed to. This is why software engineers have to consider every little detail when coding software.

If you want to learn more about building games with commands, there's another tutorial to build a clicking game which you can check out if you enjoyed this activity.
Mark as complete

Activity 3: Create Your Own Game

Introduction

Now you've created a game with the help of a tutorial, it's time to use your creativity and design your own game. Software engineers need to use their imagination to create the design and functions of the software they're developing. They create an image of how the software should look and run in their head, then produce that using codes and commands.

Activity

For this activity, you'll use the Scratch 'Create' page to build your own game. Feel free to find some inspiration on the 'Ideas' page but use your knowledge of commands to create a new game without instructions.

It can be a sports game, a classic arcade style game or any other style of game you're interested in! Once you're finished, come back to continue.

[Activity Link](https://scratch.mit.edu/projects/editor/?tutorial=getStarted) <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

Conclusion

The commands that go into software development is far more advanced than the commands needed to create a simple game, but the premise is always the same – write a command to complete a function.

Coding is a technical skill that you'll learn and develop if you decide to follow a career path as a software engineer, but having a prior understanding is always handy. So, if this is something you're interested in pursuing, we'd recommend learning and experimenting with different programming languages, so you'll be in high demand as a developer.