Beginning programming guides

# Pedagogy

Back in the 1980’s I have fond memories of typing in programs from computer magazines, to see what they would do. Inevitably, at 10 years old, I wasn’t very accurate copying the listings and there were errors. I had to study the program carefully, and correct my mistakes.

The programs would eventually run, and I would be delighted that the computer did something it couldn’t do before. The programs were very simple, so I began experimenting with changing a few variables (not that I understood what they were back in the day) to see what would happen. This intrigued me, and having picked up the “Programmers guide” and “Programming keywords” for Locomotive Basic that had come with my computer, I began to experiment with adding in extra little bits to the code, to see what would happen. Gradually I became more confident. I began to want more knowledge of keywords, to solve a particular problem I had. Using the programming keywords as a handy reference I learned the commands I could use, their syntax, and I gradually memorised them. Programming was challenging and fun.

As I became more confident I began setting myself bigger challenges, writing code from scratch, and learning more through the process. I began to learn of more appropriate techniques, and my programs became more sophisticated. As I learned new languages, I would begin to read programming books, but skip much of the text to look at the code examples, transitioning my knowledge of keywords from one language to another.

It is this learning journey to becoming a confident programmer I want to share with my students:

* Typing up code, seeing what happens: the confidence from not starting from a blank screen.
* Modifying the code to do something different: beginning to understand what the code means.
* Attempting challenges using only the commands learned: applying new knowledge gained.
* Beginning to write bigger programs with less support: becoming independent.

These guides facilitate that journey that is a good starting point for the basics of algorithms before getting into event driven and object-oriented approaches.

## How to use these resources

It is suggested that students complete an objective of the learning tasks in the language they are learning, and before moving on to the next objective, have a go at some of the challenges to support the learning tasks. The solutions provided are indicative of an approach to solving the problem, but they are not necessarily the only way of solving the problem.

Many of the newer versions of languages provide commands that can easily simplify some of the challenges. Students should be reminded that need to know the core algorithms underpinning the subject of Computer Science and therefore should only use the commands presented in the objective, and all the previous objectives to that point to solve the challenge. This will be the best preparation for answering exam questions.

## Codes used

In the code, a continuation of the same line is indicated with an arrow symbol: →
The difficulty of a challenge is graded by 🛠 icons.

Enjoy!

Craig & Dave