Algorithms and Programming

LEVELS 7-8

Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors (ACTDIP029)

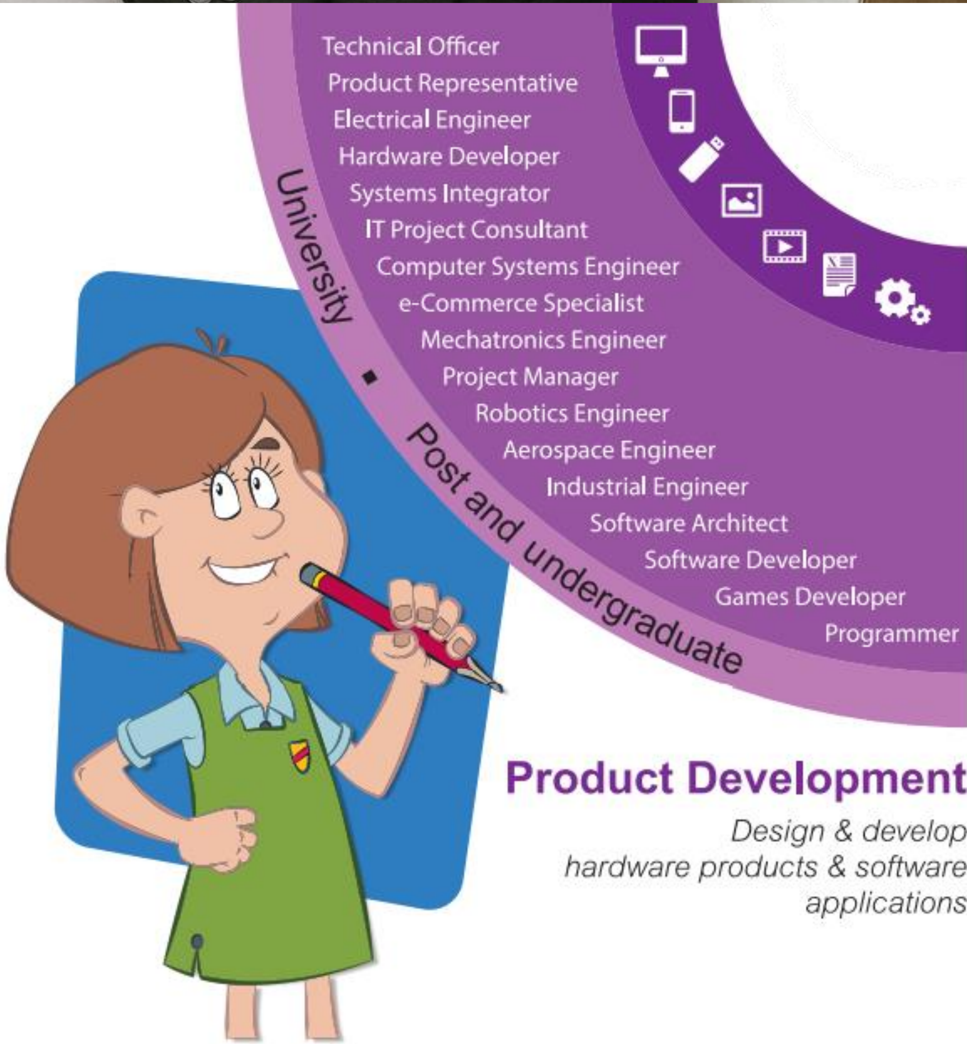
Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language (ACTDIP030)

Career Pathways

Software Engineer
Applies knowledge and skills in software to solve problems by architecting, creating and maintaining applications and systems.

Software Engineer/Software Developer
A person who applies knowledge and skills in software to both architect and design a system to meet client requirements, and also builds the application through writing code.

Developer/Programmer
Creates software and applications by writing code. Can also involve gathering requirements , designing user interfaces, writing documentation and other processes.



University • Post and undergraduate

Technical Officer
Product Representative
Electrical Engineer
Hardware Developer
Systems Integrator
IT Project Consultant
Computer Systems Engineer
e-Commerce Specialist
Mechatronics Engineer
Project Manager
Robotics Engineer
Aerospace Engineer
Industrial Engineer
Software Architect
Software Developer
Games Developer
Programmer

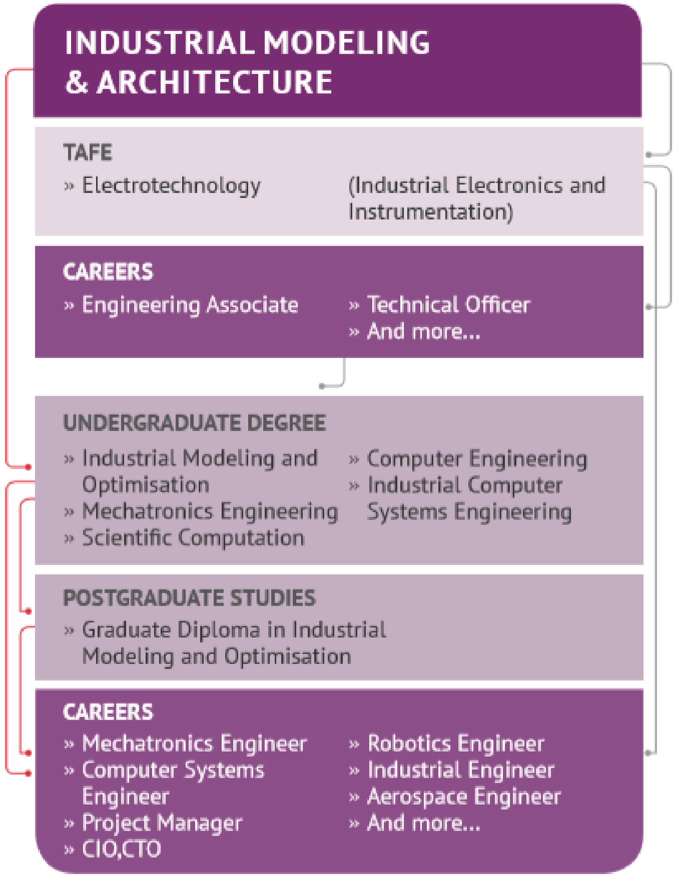
Product Development

Design & develop hardware products & software applications

Career Pathways

Video Source: Dr Tim Kitchen, filmed at the BiG Day In

Career Pathways



INDUSTRIAL MODELING & ARCHITECTURE

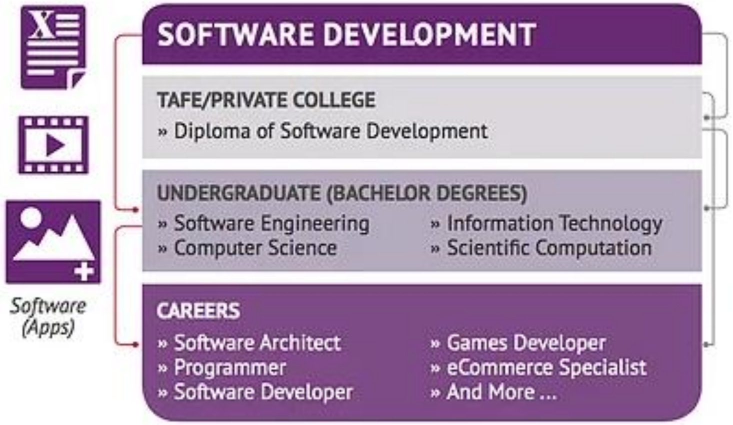
TAFE
- Electrotechnology (Industrial Electronics and Instrumentation)

CAREERS
- Engineering Associate
- Technical Officer
- And more...

UNDERGRADUATE DEGREE
- Industrial Modeling and Optimisation
- Mechatronics Engineering
- Scientific Computation
- Computer Engineering
- Industrial Computer Systems Engineering

POSTGRADUATE STUDIES
- Graduate Diploma in Industrial Modeling and Optimisation

CAREERS
- Mechatronics Engineer
- Computer Systems Engineer
- Project Manager
- CIO,CTO
- Robotics Engineer
- Industrial Engineer
- Aerospace Engineer
- And more...



SOFTWARE DEVELOPMENT

TAFE/PRIVATE COLLEGE
- Diploma of Software Development

UNDERGRADUATE (BACHELOR DEGREES)
- Software Engineering
- Computer Science
- Information Technology
- Scientific Computation

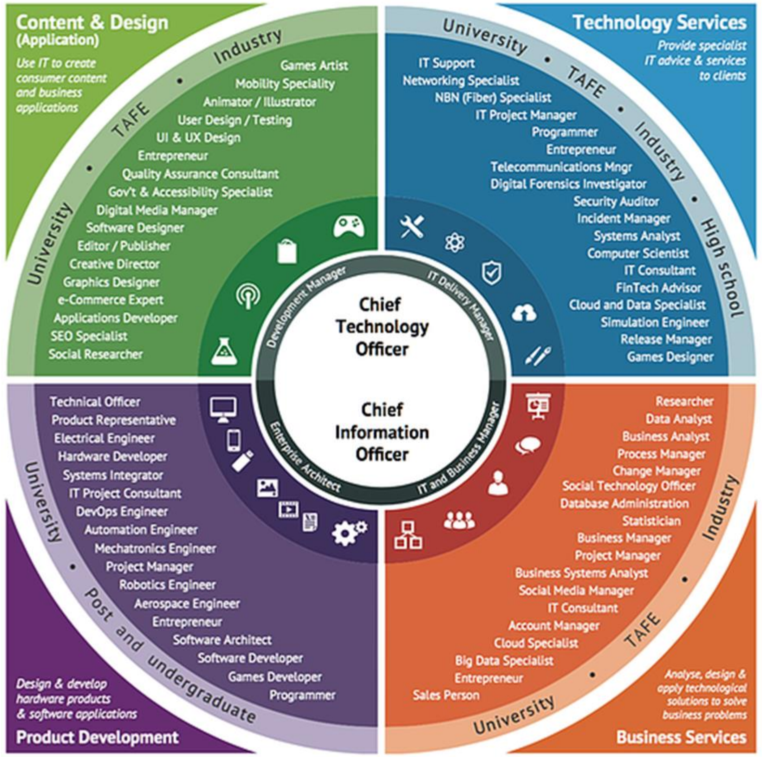
CAREERS
- Software Architect
- Programmer
- Software Developer
- Games Developer
- eCommerce Specialist
- And More ...

Pathways for Product Development Jobs

Use your coding and engineering skills to create software and hardware products. Typical Product Development roles include the design and manufacturing of hardware devices including mobile phones, tablets, laptops, computers, robots, drones and other automated products. Product Development also includes creation of the software that runs on these devices including operating systems, licensed software and applications.

The Jobs of the Future

ICT Career Wheel *for students*



Content & Design
(Application)

Use IT to create consumer content and business applications

Industry • TAFE • University

Games Artist
Mobility Speciality
Animator / Illustrator
User Design / Testing
UI & UX Design
Entrepreneur
Quality Assurance Consultant
Gov't & Accessibility Specialist
Digital Media Manager
Software Designer
Editor / Publisher
Creative Director
Graphics Designer
e-Commerce Expert
Applications Developer
SEO Specialist
Social Researcher

Technology Services

Provide specialist IT advice & services to clients

University • TAFE • Industry • High school

IT Support
Networking Specialist
NBN (Fiber) Specialist
IT Project Manager
Programmer
Entrepreneur
Telecommunications Mngr
Digital Forensics Investigator
Security Auditor
Incident Manager
Systems Analyst
Computer Scientist
IT Consultant
FinTech Advisor
Cloud and Data Specialist
Simulation Engineer
Release Manager
Games Designer

Development Manager
IT Delivery Manager
Chief Technology Officer
Chief Information Officer
Enterprise Architect
IT and Business Manager

Technical Officer
Product Representative
Electrical Engineer
Hardware Developer
Systems Integrator
IT Project Consultant
DevOps Engineer
Automation Engineer
Mechatronics Engineer
Project Manager
Robotics Engineer
Aerospace Engineer
Entrepreneur
Software Architect
Software Developer
Games Developer
Programmer

University • Post and undergraduate

Design & develop hardware products & software applications

Product Development

Researcher
Data Analyst
Business Analyst
Process Manager
Change Manager
Social Technology Officer
Database Administration
Statistician
Business Manager
Project Manager
Business Systems Analyst
Social Media Manager
IT Consultant
Account Manager
Cloud Specialist
Big Data Specialist
Entrepreneur
Sales Person

Industry • TAFE • University

Analyse, design & apply technological solutions to solve business problems

Business Services

High School
+
Tafe/Uni
+
Work Experience
=
Dream Job

The **secret ingredient** for landing your dream job is 'work experience'. When study is combined and integrated with an industry placement (such as an internship), more possibilities open up.

Definitions

Algorithm	A sequence or set of rules written as part of the code upon which software or processes are built
Branching	An instruction used in coding to perform different actions when a decision is made
Computational Thinking	Applying a four-step problem-solving cycle to develop and test solutions
Data Set	A collection of data, often displayed in a table
Data structures	A structured collection of data, often in table format (column = variable, row = a record of that variable)
Digital solutions	Creating a soluton to a problem using a digital method like an app or computer program
Flowchart	Method used to visually show the commands in your code using symbols to represent different functions in the program
General-purpose Programming Language	A programming language that can be used for writing software in a wide variety of application domains
Iteration	An instruction used in coding to repeat a command
Loop	Code that runs a set of commands either a certain number of times or until a condition is met
Pseudocode	Commands that are written in plain language (eg English) and used to plan out code and describe each step
Test data	To ensure the code produces the correct output
User interface	The digital screen through which a user inputs information such as text or clicking buttons

Computational Thinking

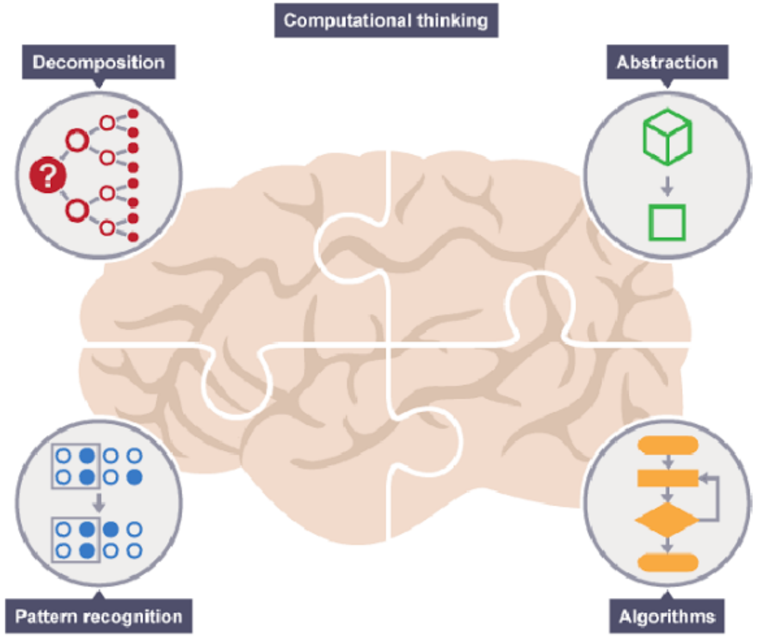
Computational thinking is a process in which you apply a four-step problem-solving cycle to ideas and challenges to develop and test solutions.

Decomposition - breaking down a complex problem or system into smaller, more manageable parts

Pattern Recognition – looking for similarities among and within problems

Abstraction – focusing on the important information only, ignoring irrelevant detail

Algorithms - developing a step-by-step solution to the problem, or the rules to follow to solve the problem



Computational thinking involves taking that complex problem and breaking it down into a series of small, more manageable problems **(decomposition)**. Each of these smaller problems can then be looked at individually, considering how similar problems have been solved previously **(pattern recognition)** and focusing only on the important details, while ignoring irrelevant information **(abstraction)**. Next, simple steps or rules to solve each of the smaller problems can be designed **(algorithms)**.

Finally, these simple steps or rules are used to **program** a computer to help solve the complex problem in the best way.

Source: Digital Technologies Hub, https://www.digitaltechnologieshub.edu.au/

Algorithms

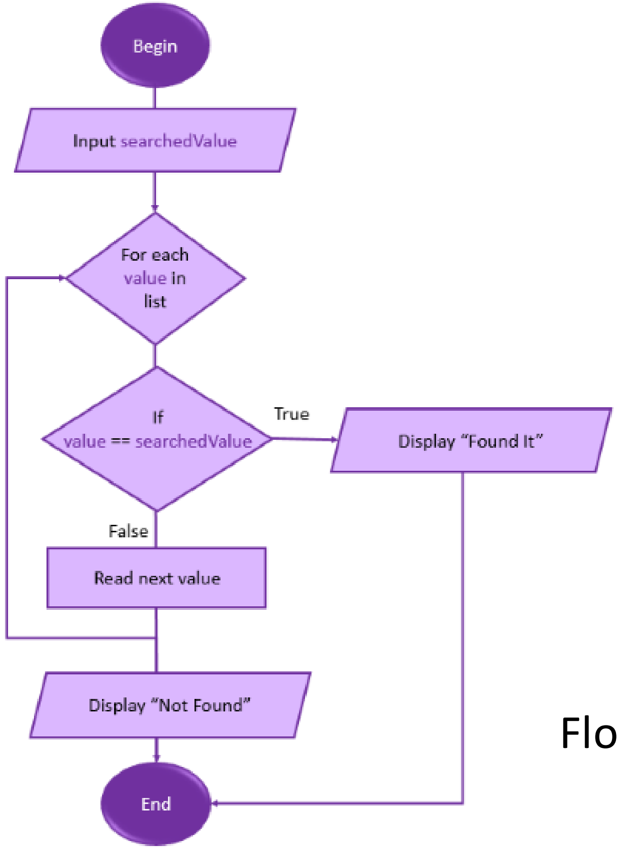
An algorithm is a sequence of explicit, step by step instructions to complete a task. Algorithms enable digital systems to complete required tasks. The complexity of algorithms increases as the task becomes more complicated. Certain functions can be used to reduce the amount of content written to perform the algorithm.

Written algorithms can be manipulated and added to a digital software program to create a digital solution. These algorithms need to be written in a language the digital system will understand. General purpose programs that have the ability of the create an array of different problems to solve. Common general-purpose programming languages include C++, Python and JavaScript, but there are hundreds of programming languages.

Video Source: Intro to Algorithms, CrashCourse

Common Algorithms

Search - step-by-step procedure deployed to locate specific data within a dataset



Flowchart

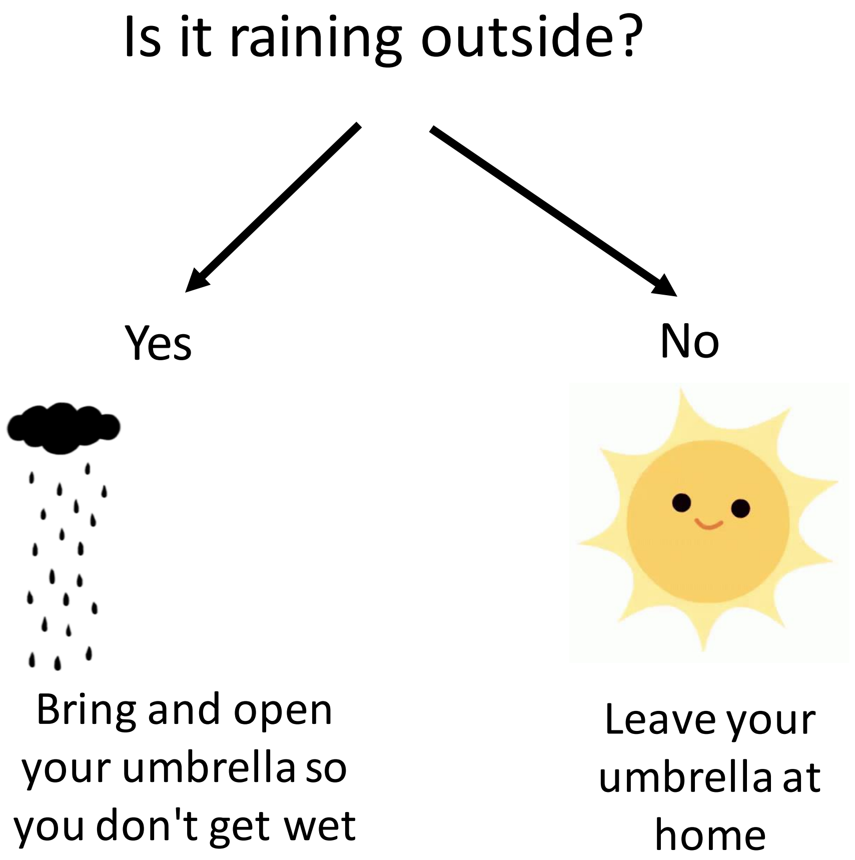
Selection Sort – sort a list or random number dataset (array) from smallest to largest

```
selectionSort(array, size)
  repeat (size - 1) times
  set the first unsorted element as the minimum
  for each of the unsorted elements
    if element < currentMinimum
      set element as new minimum
  swap minimum with first unsorted position
end selectionSort
```

Pseudocode

ALGORITHMS – Branching

Branching is the term given to show multiple options available for the task to be completed. The direction of the algorithm will change, depending on how the task is executed. E.g. The question 'Is it raining outside?' requires two options - an answer of yes or no. The answer to the question will influence the next command, to bring an umbrella or not.



Is it raining outside?

Yes

No

Bring and open your umbrella so you don't get wet

Leave your umbrella at home

ALGORITHMS – Iteration

Some instructions will be repeated, just like the instructions of making your breakfast every morning. Instead of repeating the same algorithm over and over – one algorithm can be written to repeat the same task. This is formally called an **iteration –** more commonly known as **looping**.

Instead of writing out the instructions to make breakfast everyday (365 times) you write down the instructions once, and keep using those same instructions.

Video Source: Code.org, Hour of Code, Chris Bosh teaches Repeat Until statements

ALGORITHMS – Conventional Statements

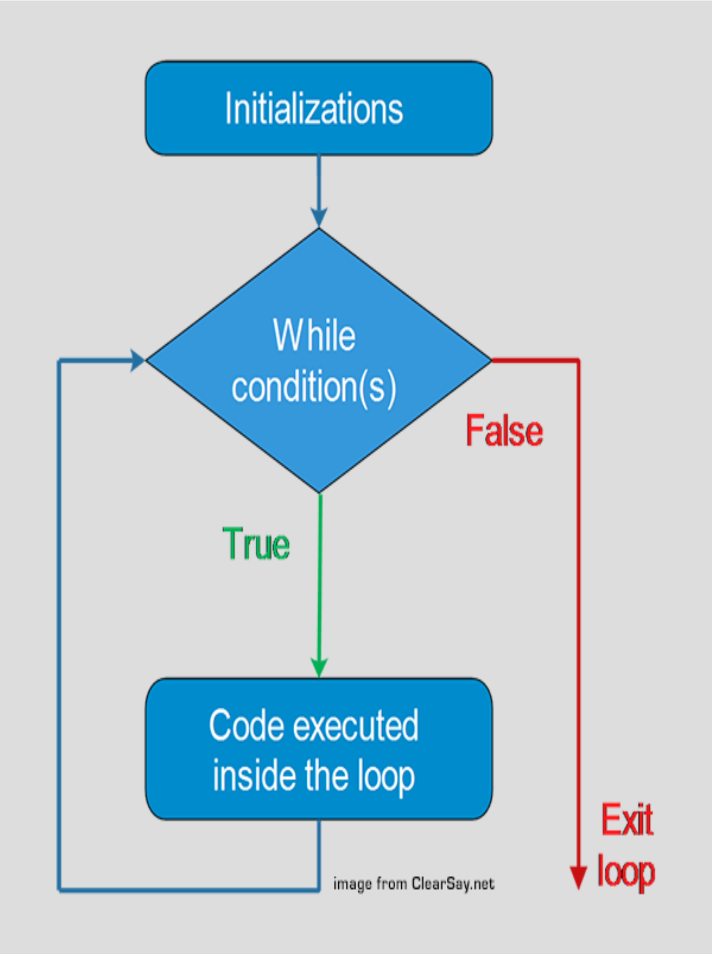
There are different ways to write code. What you use will depend on the type of program you want to create and the purpose of that program. When coding always ask yourself – are there commands I can use to reduce my code? A While loop is a good example of efficient coding:

While loop Repeat a section of code an unknown number of times until a specific condition is met. A while loop will run forever until the condition is met

Condition – Is it raining?

No (condition false) Go to the shops (avoid the statements inside the while loop)
Yes (condition true) It is raining so I will stay inside and keep checking to see when the rain has stopped.

When it has stopped, I'll go to the shops



Initializations

While condition(s)

True

False

Code executed inside the loop

Exit loop

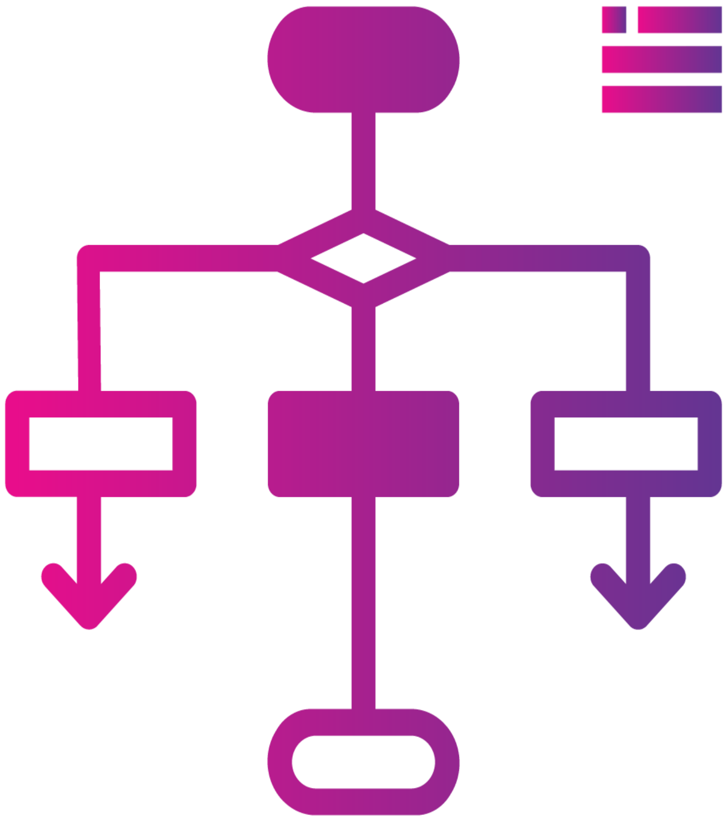
image from ClearSay.net

PLANNING TO CODE

Planning out your code is an essential step to programming.

If we go straight to the code, we may miss steps, forget code or add the wrong code. If one thing is missing from our code, it means our program may not function at all or may not perform the task we wanted. Trying to find the mistakes can be a hard task, especially if we don’t know what the actual error is because we never planned out the code.

If we do not plan before we code, we end up focusing only on the coding, not the logic steps we need to make our program complete the task. A flowchart helps plan out the code needed to create a program. Each shape has a different meaning, and arrows are used to shows the next step in the sequence depending on the input.



Flowcharts & Diagrams

We have different symbols that are used in the English language to mean different things. A full stop shows at the end of sentence. A ‘?’ is used to ask a question. Symbols are also used to show/represent different functions in a flowchart.

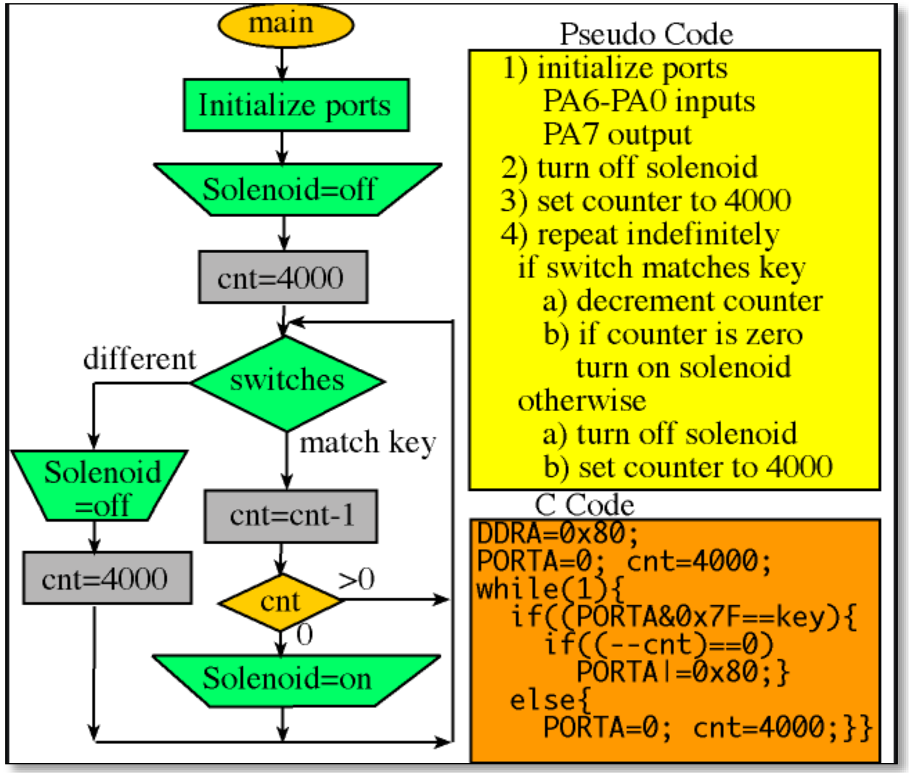
Start/End	▢	Begins and ends the functions and instructions
Flow/arrow	→	Shows the movement and direction of the flow chart
Process	▭	Shows the instructions
Decision	◇	Branches out to show the different options
Data	▱	

Video Source: CSER, An overview of flowcharts

Flowcharts & Diagrams

Flowcharts use visual cues to show the steps and commands your program will run. Flowcharts are important to order your commands. You can go back, follow the commands and make sure it flows. Flowcharts will help you track your progress when programming and before you code – they help you understand all the different functions that will be needed to create your program. We can use the flowchart and pseudo code to remove any sequencing errors that may occur, saving time and frustration when writing the code.

Pseudocode is another way to plan before you code. Pseudocode is a language-based description of the steps in an algorithm or system. It mostly uses structural conventions of a normal programming language; however, it is intended for human reading. It doesn't include programming language-specific code. It is a description of what happens in each step and includes important information needed for the algorithm or program to succeed.



Pseudo Code

```
1) initialize ports
     PA6-PA0 inputs
     PA7 output
2) turn off solenoid
3) set counter to 4000
4) repeat indefinitely
   if switch matches key
     a) decrement counter
     b) if counter is zero
          turn on solenoid
   otherwise
     a) turn off solenoid
     b) set counter to 4000
```

C Code

```
DDRA=0x80;
PORTA=0; cnt=4000;
while(1){
  if((PORTA&0x7F==key){
    if((--cnt)==0)
      PORTA|=0x80;}
  else{
    PORTA=0; cnt=4000;}}
```

An algorithm flowchart, pseudocode and code

General Purpose Programming Languages

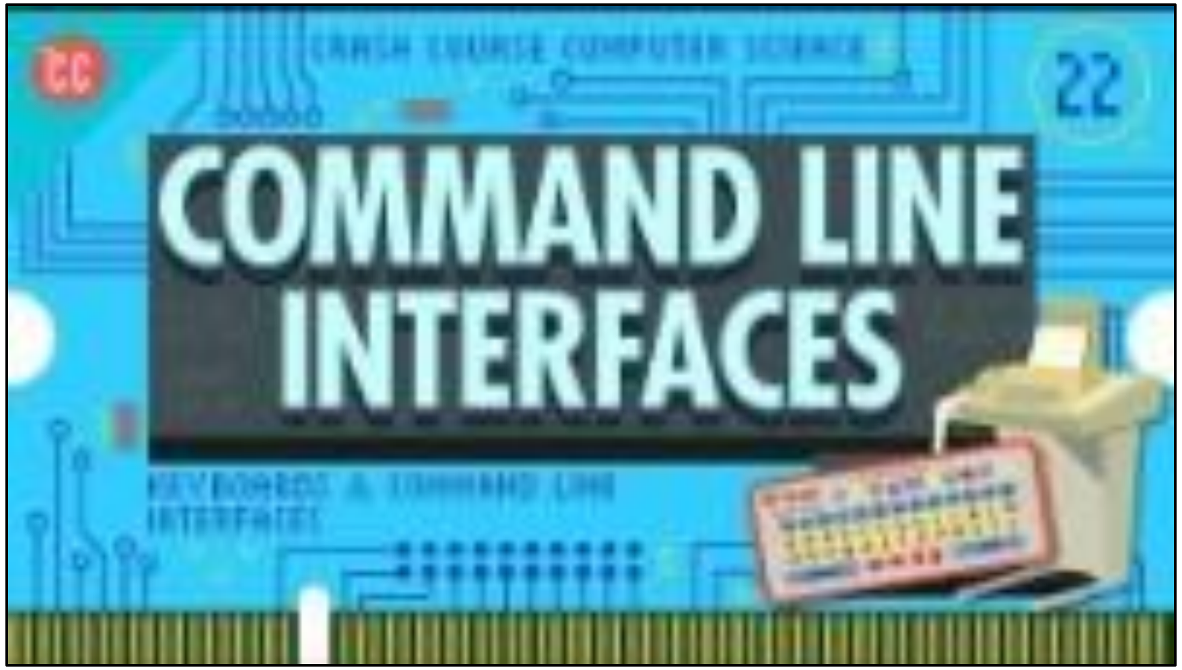
After you have General purpose programs that have the ability of the create an array of different problems to solve. Common general-purpose programming languages include C++, Python and JavaScript.

Video Source: Simplilearn YouTube Channel

User Interface

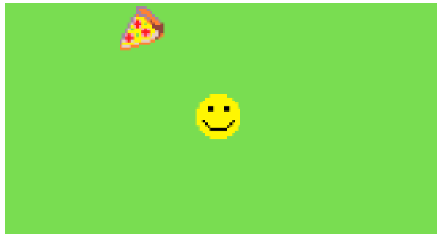
User interface refers to how the user will interact with the hardware and software. Interacting with the software may include creating software with a Graphical User Interface (GUI) which often includes creating buttons to add text, menu buttons with multiple options and pages that when clicked will access more pages. Any part of the program that the user interacts with is part of the user interface.

Command Line Interface is a common interface when programming. This is when the user will type in the command. The command is displayed in the form of text (in the programming language not structured English). The digital device will respond to the commands.


CC
CRASH COURSE COMPUTER SCIENCE
22
COMMAND LINE INTERFACES
KEYBOARDS & COMMAND LINE INTERFACES

Video Source: CrashCourse, Keyboards & Command Line Interfaces

Develop a Digital Game

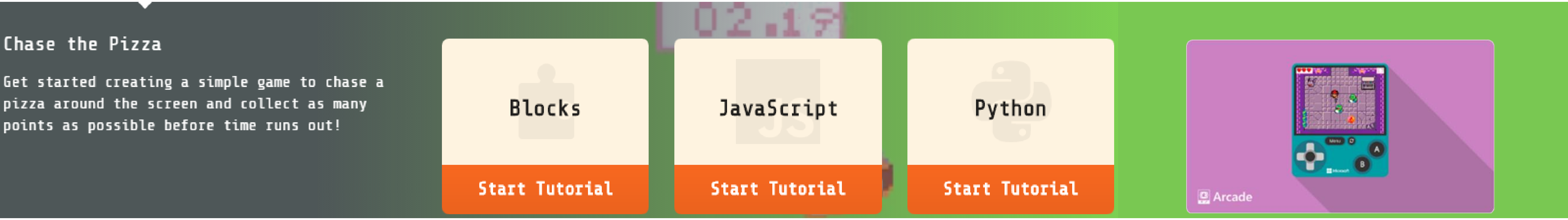


Chase the Pizza

Making a digital game is fun and creative. It also uses many basics of coding.

Online tutorials can help us create digital games. Once we learn some basics, we can make our own games. Microsoft MakeCode is an online app that teaches you how to make games and computer programs.

Go to https://arcade.makecode.com/ and choose a game tutorial to complete. Try using the Blocks Tutorial first, then create another game using JavaScript or Python.



Chase the Pizza

Get started creating a simple game to chase a pizza around the screen and collect as many points as possible before time runs out!

Technical Advisor

David Beitey *is an IT Industry Professional with a Bachelor of Information Technology (Honours). David is currently the Online Technologies Manager at the eResearch Centre at James Cook University. His experience lies in full stack web and application development, user experience, systems administration and integration, project management and business analysis.*

David is an organiser of the DevNQ (Developers North Queensland) community and active open source software developer. He regularly mentors students, researchers and colleagues, and has previously been involved in the CSIRO STEM Professionals in Schools, a national volunteer program that partners schools and industry to bring STEM (Science, Technology, Engineering and Maths) experiences into the classroom.

David's expertise and oversight has guided the publication of the resource and the ACS thanks David for his partnership in the ICT Gateway to Industry Schools program.

Acknowledgements

About ACS

ACS is the professional association for Australia's technology sector. More than 48,000 ACS members work in business, education, government and the community. ACS exists to create the environment and provide the opportunities for members and partners to succeed.

ACS strives for technology professionals to be recognised as drivers of innovation in our society, relevant across all sectors, and to promote the formulation of effective policies on technology and related matters. Visit www.acs.org.au for more information.

About the ICT GISP

The Information and Communications Technology Gateway to Industry Schools program encourages partnerships between industry, government, schools and their communities to build Queensland's future information technology workforce. The program provides an important opportunity to address the significant shortfall of young, emerging ICT talent in Queensland. Access more information and ICT teaching resources below:

ICT GISP Website - https://qldictgisp.acs.org.au/home.html

ICT Educators Community of Practice - https://www.acs.org.au/ict-educators.html

The Big Day In ICT Careers - https://www.thebigdayin.com.au/

ICT Careers Wheel - https://qldictgisp.acs.org.au/career-pathways.html

The Department of Employment, Small Business and Training funds this Gateway to Industry Schools Program initiative

Acknowledgements

The ACS ICT Educators resources are licensed under a Creative Commons Attribution 4.0 International License.

Schools and educators can join the ICT Educators’ Community of Practice and can, without charge:

- Use — Use for own purposes.
- Share — Copy and redistribute the material in any medium or format.
- Adapt — Remix, transform, and build upon the material for any purpose, even commercially.

When sharing, or adapting, you must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

All users are invited and welcome to collaborate with ACS and other users to create and maintain the resources, and participation will be acknowledged.
More information about creative commons can via the creative commons website:
https://creativecommons.org/