1953
Invention of High Level Programming Languages

Learning Outcomes addressed in this section are listed below.

1.11	discuss the complex relationship between computing technologies and society including issues of ethics
1.12	compare the positive and negative impacts of computing on culture and society
1.13	identify important computing developments that have taken place in the last 100 years and consider emerging trends that could shape future computing technologies
1.18	recognise the diverse roles and careers that use computing technologies
1.22	read, write, test, and modify computer programs
1.23	reflect and communicate on the design and development process
2.4	illustrate examples of abstract models
2.5	use pseudo code to outline the functionality of an algorithm
2.6	construct algorithms using appropriate sequences, selections / conditionals, loops and operators to solve a range of problems, to fulfil a specific requirement
2.7	implement algorithms using a programming language to solve a range of problems
2.17	use ASCII and Unicode character sets to encode/decode a message and consider the importance of having such standards

When other Learning Outcomes are addressed, for instance in classroom activities or through related online resources, the LO is numbered.

Grace Hopper and the UNIVAC

In the early years of computing technology, electronic computer could only be programmed by numbers, tape, punch cards or even manually manipulating the thermionic valves (vacuum tubes) to certain settings. After World War II, Grace Hopper worked on the first commercial computer called the UNIVAC. In 1953 she invented the first high level programming language, A-0, that used words and expressions to program the UNIVAC. She also created the first modern day compiler and coined the phrase BUG. A-0 evolved into Flow-matic and eventually became the widely used COBOL. Here are some lines of COBOL:

```
PROCEDURE DIVISION.
        LCCS-FIRST-PARA.
        DISPLAY ‘Hello World.’.
        MOVE ‘Grace Hopper’ TO WS-NAME.
        DISPLAY ‘My name is: ’ WS-NAME.
STOP RUN.
```

The output will display

```
Hello World.
My name is: Grace Hopper
```

In 1954 IBM designed FORTRAN (FORmula TRANslation) which becomes an industry standard, especially in the area of scientific analysis. An IBM team led by John Backus developed the language which is still in use today. Here are some lines of FORTRAN:

```
DO
        a = SQRT(b) + c
        IF (a > z) THEN
                !// Exit the loop (comment)
                EXIT
        END IF
END DO
```

Watch 2 jargon buster videos from a 1983 TV series Bits and Bytes.

1.What is the difference between a compiler and an interpreter?

2.Explanation of High Level Languages.

What would you add to the list at the end of the languages video?

LO 2.4

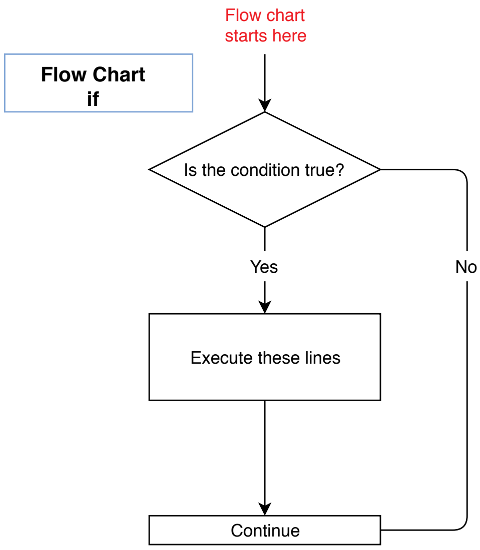
From
Flow-matic (1950s)
To
FORTRAN (1950s)
To
BASIC (1964)
To
LOGO “Turtle” (1967)
To
Pascal and C (1970s)
To
Perl and Mathematica (1980s)
To
Java, Javascript, Swift, PHP, Python and Scratch (1990s and beyond)

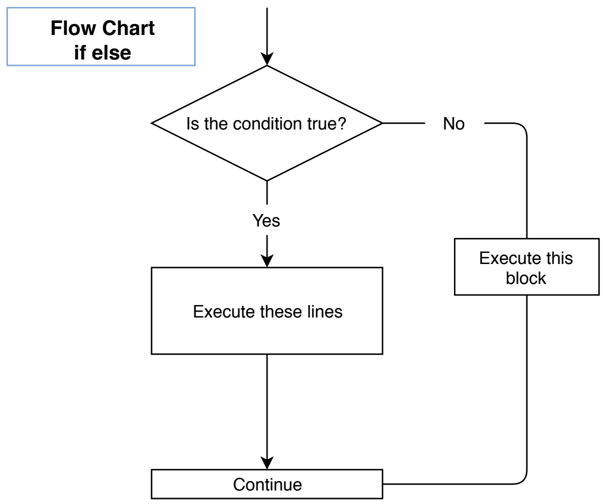
Explore the timeline above.

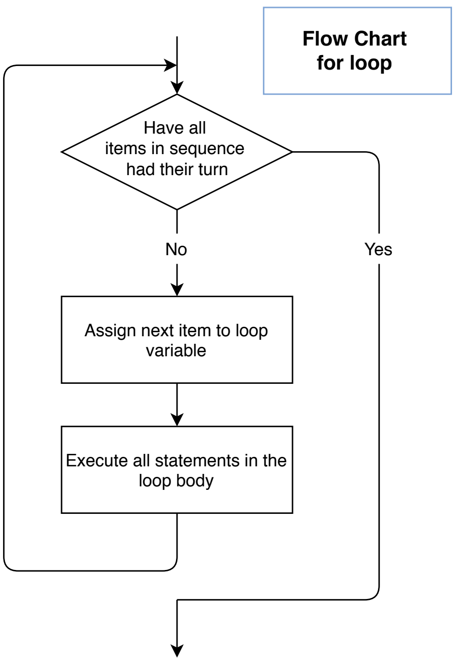
Watch a video on getting Y2K ready – BUG hysteria or good preparation on NY eve 1999?

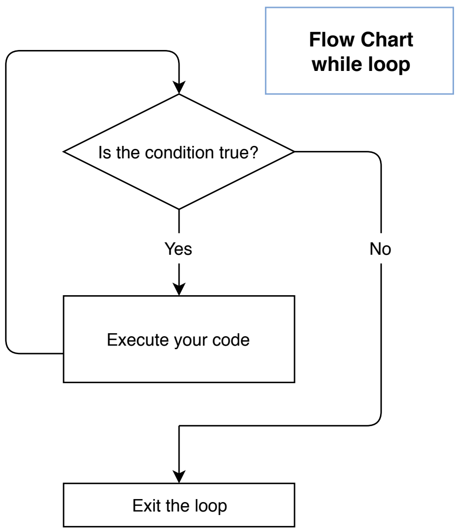
LO 1.11, 1.12

This example shows many similarities to the structure of most high level languages and scripting languages such as Python and Javascript. One of the features of Grace Hopper's flowmatic programming language was a conditional called an IF statement. The flow charts below illustrate decision-making algorithms now common place in high level languages: conditionals and loops.









In 1964, BASIC (Beginners All purpose Symbolic Instruction Code) was invented by two Maths professors, Kurtz and Kemeny. They designed their language specifically for learners. A decade or so later it was embedded on the earliest forms of personal computers mainly due to its adoption by a then little-known start-up called Microsoft. An example of the Hello World program in BASIC is below. (REM signifies a comment.)

```
10 REM Hello World in BASIC
20 PRINT "Hello World!"
```

The Hello World website demonstrates the introductory Hello World program in almost 600 programming languages.

To explore in further detail, the CS Field Guide to Programming Languages is an interactive, uncomplicated yet comprehensive introduction to this area, with some insightful interactive examples on low level languages, including machine code, and also interpreted languages such as Python. The Guide also explores ASCII and Unicode (UTF-8, UTF-16, UTF-32) in depth, including exercises on converting text to ASCII and Unicode, the limitations of using only 7 of the 8 bits in ASCII, and how to represent the word for cat, in Chinese. (LO 2.17)

High level languages are an abstraction from lower level languages. The programmer does not need to know what is going on below the level of the language, never mind at the lowest level of ones and zeroes. In this sense it seemed inevitable that programmers would need to be facilitated in writing programs that resembled spoken languages more than they resembled machine code.

The **innovation to use a keyboard** to enter data and construct programs, revolutionised the world of computing, as far back as 1956. "At MIT, researchers begin experimenting with direct keyboard input to computers, a precursor to today´s normal mode of operation. Typically, computer users of the time fed their programs into a computer using punched cards or paper tape. Doug Ross wrote a memo advocating direct access in February. Ross contended that a Flexowriter – an electrically-controlled typewriter – connected to an MIT computer could function as a keyboard input device due to its low cost and flexibility. An experiment conducted five months later on the MIT Whirlwind computer confirmed how useful and convenient a keyboard input device could be." [7]

Is it a matter of time before a virtual, or laser, keyboard becomes the normal means of keyboard interface? It is very interesting to discuss the impact of keyboard input on the world of computers and the advance of computing technologies. Is it an exaggeration to say that keyboard input has revolutionised our world and democratised the world of computer programming?

Take the Crash Course on programming languages

From the Crash Course Series on YouTube (Carri Anne Philbin)

LO 1.18
LO 2.4

There are more resources and exercises on ncca.ie related to this area.

Download the CT challenge, which uses standard ASCII/uft-8 code to implement a basic Caesar shift encryption in Python, housed within a html file. Note that the head of the html file refers explicitly to the utf-8 character set.

LO 1.22, 1.23
LO 2.5–2.7, 2.19

Discuss the impact of keyboard input on the roles and careers in computers. How has it helped to spread digital literacy? Can you see a world where there is direct input to a computer, making the keyboard redundant?

Facilitate a *Walking Debate* on the above questions and/or *TPSS* activity on this area.

LO 1.13, 1.18

7 computerhistory.org
http://www.computerhistory.org/timeline/computers/#169ebbe2ad45559efbc6eb35720ba5f3

The journey of the letter Q from the keyboard to processing is many ways the same now as it was in 1956. But only at an abstract level. Follow the journey below from the moment you press the letter Q on your keyboard to its apparently instantaneous appearance on your screen. (Well most of the time!)

Join the Q!

1. YOU PRESS Q

The keyboard sends a signal (scancode) to the keyboard controller saying a button has been pressed.

2. PATIENT KEYBOARD CONTROLLER

Reads the scancode, sees the letter Q and stores it until the processor is ready.

3. IMPATIENT KEYBOARD CONTROLLER

Alerts the computer's processor (an interrupt) that it has input.

4. BUSY PROCESSOR

You can imagine the processor is quite busy. When it is ready to deal with a keyboard interrupt, it alerts the Operating System (OS)

5. MULTI - TASKING OS

The OS locates the window you were working in when you pressed the letter Q.

6. UPDATE WITH YOUR LETTER

Let's say, for example, you are in a word processing application, like notepad or MS Word. It adds Q to the working area of your file.

7. 1 BYTE OF THE Q

Q takes up 1 byte of memory. (8 bits of Unicode or 7 (used) bits of ASCII)

8. LET THE OS KNOW

The window tells the OS to display Q. The OS obliges by adding it to the video card memory.

9. ON THE NEXT REFRESH, Q APPEARS

Monitors tend to be refreshed over 60 times per second.

YOU PRESS

YOU SEE

Computer science has created careers, roles and employment for people that we could never have imagined. A browse on the web of the most sought after skills in the digital hi-tech industry will most likely include: Software Engineer, Cloud Architect, CS Analyst, Data Scientist and Web Developer. Picture yourself in 1977, at the launch of Apple II, arguably the first modern PC. Could even visionaries such as Steve Wozniak and Steve Jobs have foreseen that careers in the CS industry would include Web Developers or Cloud Architects? With neither a spider nor a meteorologist in sight! However the first decades of the 21st century saw the rapid expansion of AI and cloud computing. It prompted many thinkers and leaders such as US President Barrack Obama to question whether the digital revolution will be the first industrial revolution in history to be a net destructor of jobs as opposed to a net creator of jobs.

The World Economic Forum did a major report on The Future of Jobs.

Review their findings on employment trends and the disruptive effect of new technologies.

LO 1.18

Stimulate a Debate on how computer science is changing our world.

The model for this activity is explained in A Summary of Teaching & Facilitation Methodologies.

1. Watch a Stimulus Video or read a stimulus piece.

- How CS is changing our world is a video overview of new careers and opportunities opening up.
- The first 8 minutes of the BBC video Will Robots Take Our Jobs, examines the impact of automated ports, focusing on Rotterdam. The remaining 20 minutes discusses the wider impact of CS on our world of roles and careers.
- The following piece is from Michael Dertouzos, speculating on the impact of CS on our world in the future. It was 1995, and he had just come from a meeting in MIT, chaired by the Tim Berners-Lee, the inventor of the world wide web.

“In a quiet but relentless way, information technology would soon change the world so profoundly that the movement would claim its place in history as a socioeconomic revolution equal in scale and impact to the two industrial revolutions. Information technology would alter how we work and play, but more important, it would revise deeper aspects of our lives and of humanity: how we receive health care, how our children learn, how the elderly remain connected to society, how governments conduct their affairs, how ethnic groups preserve their heritage, whose voices are heard, even how nations are formed. It would also present serious challenges: poor people might get poorer and sicker; criminals and insurance companies and employers might invade our bank accounts, medical files, and personal correspondence. Ultimately, the Information Revolution would even bring closer together the polarized views of technologists who worship scientific reason and humanists who worship faith in humanity. Most people had no idea that there was a tidal wave rushing toward them.” [8]

8 Dertouzos (1998) *What Will Be* HarperCollins.

LO 1.11

Students should be able to discuss the complex relationship between computing technologies and society including issues of ethics

2. Prompt questions to provoke class discussion and elicit initial viewpoints.

Rasmussen college pose 6 ways in which Computer Science improves the world we live in today. Do you agree that the evolution of computing technology has been only positive in the 6 areas below?

1. Solving problems and improving solutions
2. Protecting people and organisations
3. Furthering education
4. Improving communication
5. Organizing & streamlining philanthropy
6. Positively impacting every area of society

3. Divide into research groups to explore the topic from key standpoints.

If feasible, form at least 6 groups based on the 6 categories above. Each group examines one of the prompts above, analysing the particular standpoints below or researching the questions posed below. (The list of 6 prompts is suggestive and ideally adapted in an appropriate manner for the students)

For example, for prompt number 4. Improving communication, research the following:

- Is CS a net creator or destroyer of jobs in this area?
  (Give an overview of the type of jobs created or destroyed and justify your conclusions)
- How has CS helped to improve our world in this area?
- How has CS helped to disimprove our world in this area?

4. Choose a teaching / facilitation methodology.

Adapt appropriately for your CS classroom.

a. Students first research each topic in research groups of 3.
b. Use a *Jigsaw Learning Technique* to create groups of 3 comprising one student from 3 different themes. Each person discusses their research within their new group.
c. Reassemble into original groups.
   Each group has up to 5 minutes in the *Hot Seat*
   OR
   A *Power of Persuasion* technique is used to group students into their preferred category and try to convince other students over to their viewpoint.
   OR
   A *Think-Pair-Share-Snowball (TPSS)* exercise to broaden out the findings and conclusions of each research group.

LO 1.12

Students should be able to compare the positive and negative impacts of computing on culture and society