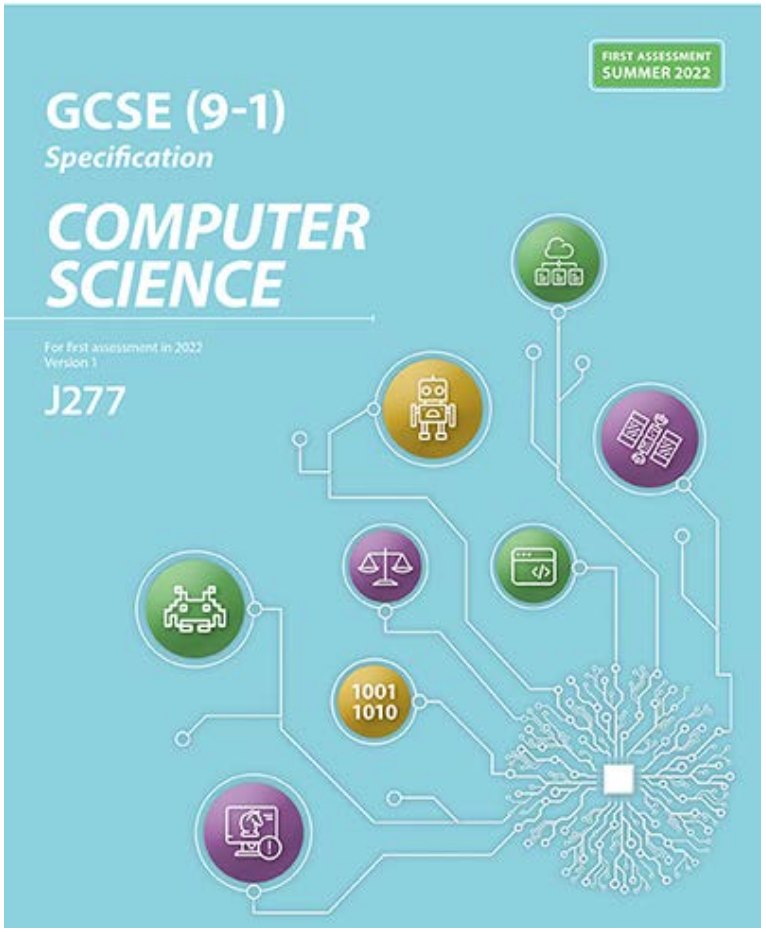
Year 9 Computer Science Transition Project



Alleyne's Academy

Introduction

- So, You have chosen to take Computer Science and are about to start the course in September…?

- Firstly, well done for making an excellent decision! Computer Science is a fantastic course, which will open up a whole new world for you. You will learn how computer systems work and will enable you to program increasingly complex programs and applications.

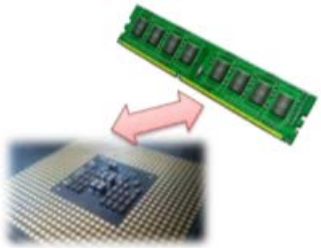
- Secondly, to make sure you feel prepared for when we start, work your way through this transition work. If you encounter any issues then feel free to email me at j.millington@alleynes.staffs.sch.uk and I will get back to you.

- Finally, don't worry if you struggle with some of this work, it is meant to be tough, it's showing you how much stuff you are going to be able to learn in Computer Science.

Section 1:
CPUs and the Fetch Execute Cycle

- The CPU is often known as the 'brain of the computer'. Its job is to process data. And by processing we mean things like searching, sorting, calculating and decision making. Whenever you are on working on your computer, it is the CPU which is at the heart of everything.

- The CPU follows three steps in order to process data. It is known as the Fetch - Decode - Execute cycle (aka Fetch-Execute Cycle). Whenever you open and work with a program, its data and instructions are loaded onto the RAM. As the RAM is accessed directly by the CPU, the CPU can get to work!

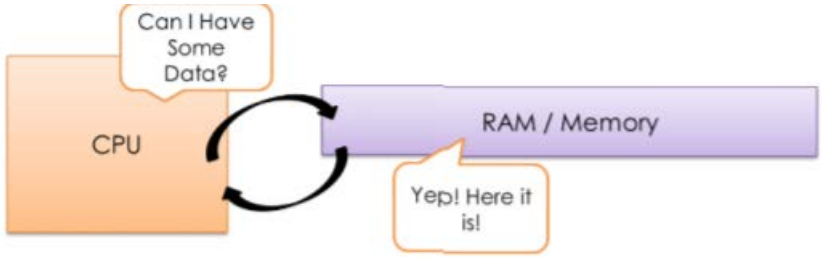


- Let’s break this down into the 3 main stages – (This is really complex stuff so just getting a basic idea at this stage is great!)

Fetch

The Fetch Stage

- In this step the CPU fetches some data and instructions from main memory (RAM) and then stores them in its own temporary memory called 'registers'. For this to happen, the CPU uses a piece of hardware called the 'address bus'.



- The address of the next item that the CPU wants is put onto the ‘address bus’.

- Requested data then travels from the RAM to the CPU on another piece of hardware called the ‘Data Bus’.

Decode and Execute

The Decode Stage

- The decode step is where the CPU understands / works out what the instruction it has just fetched actually means. The CPU 'decodes' the instruction and gets things ready for the next step.

The Execute Stage

- The Execute stage is where data processing happens. Instructions are carried out on the data. Once a cycle has been completed, another begins, and so it goes on. And on. And on and on and on.

CPU Performance

- There are several factors that can impact the CPU's performance, for now, let's just look into Clock Speed

Clock Speed

- The speed of the Fetch-Decode-Execute cycle is determined by the CPU's clock chip. This chip uses a vibrating crystal that maintains a constant rate. The speed of the clock is measured in hertz (Hz) which is the amount of cycles per second. A clock speed of 500Hz means 500 cycles per second. Current computers have CPU clock speeds of 3GHz which means 3 Billion cycles per second.

Overclocking

- It is possible to increase the clock speed for a CPU. This is known as overclocking. In theory, if the clock is faster, then the CPU can perform more calculations and therefore perform faster. The problem is that CPUs get hotter the more work they do – so overclocking is dangerous without the appropriate heat management.

If you want to push yourself, do some of your own independent research into 'CPU Cores' and 'CPU Cache'!

Section 1: Questions

1. Name the steps that the CPU carries out in order to process instructions. [3]

2. Describe what happens during each of the stages written in your last answer. [3]

3. Bob's PC struggles to perform when playing the latest computer games, due to the CPU being unable to process instructions quick enough. Instead of upgrading his CPU, what could he try to do to improve the performance of his CPU and what issues may he be faced with? [3]

Section 2: IPs and ISPs

IP Address

- This means INTERNET PROTOCAL ADDRESS. It is a unique number (e.g. 324.45.321.23) given to every computer on the internet – no two computers can have the same address. It's just like a postal address – used to identify a house – no two houses have the same address! Interestingly, a device's IP address will not necessarily remain the same each time it joins a network.

Not to be confused with ISP

ISP

- This means INTERNET SERVICE PROVIDER. This is simply the company who provide you with your internet connection.

Web-Hosting and URLs

URL

- This means UNIFORM RESOURCE LOCATOR. This is simply a fancy name for a web address, such as “http://www.bbc.co.uk” or “http://www.google.com”.

Web Hosting

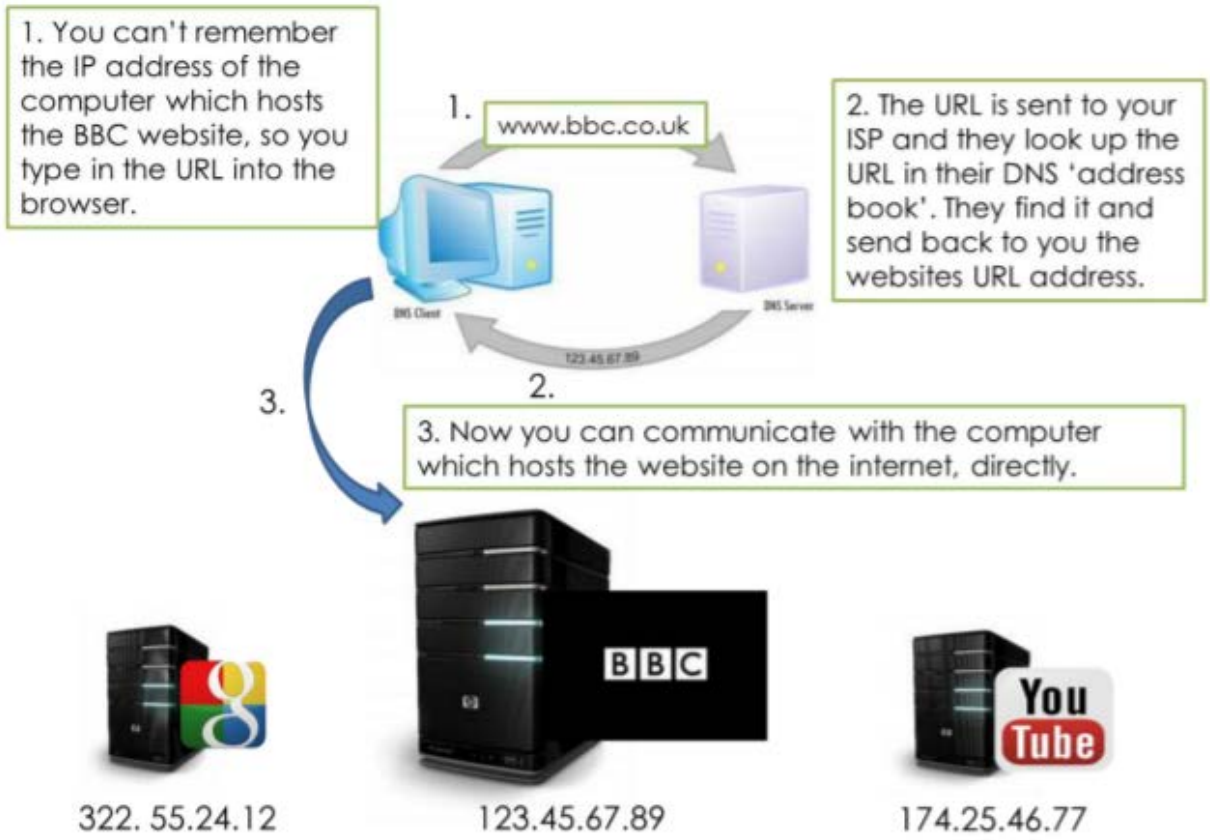
- When visiting websites, we will usually type a websites address into a browser’s address bar. However, our computer can only connect to these websites if it knows the websites IP address. So how do we get our computers to connect to websites, if we don’t know their IP addresses? Well, computers use a system called DNS and this system uses these acronyms that we have just spoken about.

DNS

DNS

- This means DOMAIN NAME SYSTEM. This is the system used to find the computer which hosts the website you are looking for

How DNS works:



Section 2: Questions

1. Explain, using examples, the following acronyms: IP address, ISP and URL. [3]

2. What is DNS for and how does it work? [5]

Extra Research – Look into different types of Network threats such as Ransomware and Brute Force Attacks

Section 3: The Operating System

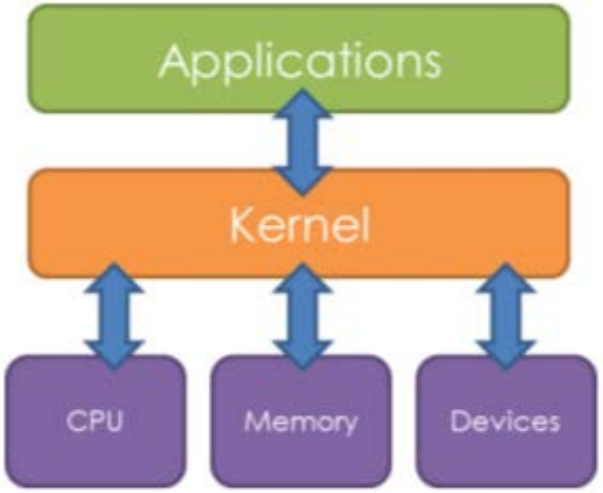
- The operating system is the most important piece of software on any computer. Without it, no programs will run. This is because an operating system is responsible for controlling / communicating with the computer hardware. It provides a platform on which games, browsers, music players etc., can all work.

- If you were to run an everyday program (e.g. a word processor), without an operating system, nothing would be displayed on the screen, nothing could be sent to the printer, nothing could be typed. This is because application software does not know how to TALK to hardware devices. However, the operating system does. The operating system can also talk to the application that is running. So when you print a document, the application talks to the OS, which in turn talks to the printer.

- The operating system sits between the user's applications and the hardware. It enables applications to use the hardware resources.

The Kernel

- The kernel is the heart of the operating system and is responsible for looking after “the most low-level hardware operations”. It is the kernel that applications make use of when they want to operate the computer’s hardware.



Section 3: Questions

1. Using an example, explain why an Operating System is essential when running applications on a computer. [4]

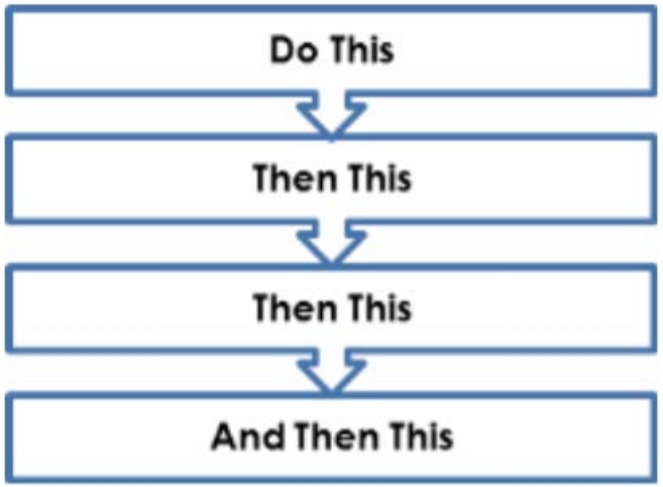
2. What is the role of the Operating System's Kernel? [2]

Section 4:
Programming Constructs

- Programming languages have a set of statements to determine how to reach a goal. The FLOW of these statements is CONTROLLED by 3 different structures:
  - Sequencing
  - Selection
  - Iteration

Sequencing

- This means performing each instruction in order, one after the other.



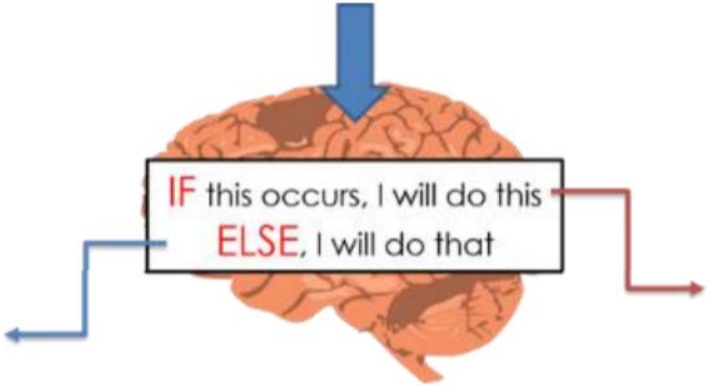

```
number1 = integer
number2 = integer
answer = integer

INPUT number1
INPUT number2
answer = number1 + number2

PRINT answer
```

Selection

- This is where the program has to make a decision and then carry out different instructions based on the answer.




```
month = input("Please enter your month number")
month = int(month)

if month == 2:
    print("Your month has 28 days")
elif (month == 4) or (month == 6) or (month == 9) or (month == 11):
    print("month has 30 days")
else:
    print("Your month has 31 day")

input()
```

Iteration

- Iteration is a posh programmer word for ‘Loop’ or ‘Repeat’.
- So this is where the program repeats a section of its code for a set number of times OR until a certain criteria / decision is reached.


```
count = 10

while count > 0:
    print("Hello Word")
    count = count - 1

input()
```

```
Hello Word
Hello Word
Hello Word
Hello Word
Hello Word
Hello Word
Hello Word
Hello Word
Hello Word
Hello Word
```

Section 4: Question

- Label ‘Sequencing’, ‘Selection’ and ‘Iteration on the following code.

```
import time

menuchoice = 0

print('*****Menu*****')
print('')
print('1. Display my name')
print('2. Display my age')
print('3. Display my address')
print('')

while True:
    try:
        while(menuchoice < 1) or (menuchoice > 3):
            menuchoice = int(input("What is your menu option?"))
        break
    except ValueError:
        print("Please type in a number not a word")


if menuchoice == 1:
    print("Mr Wickins")
elif menuchoice == 2:
    print("29 years old")
elif menuchoice == 3:
    print("Sidmouth College")

print("Goodbye")
```

- **Extra:** For extra preparation, sign up to Solo Learn and work your way through the Python tutorials!

Section 5: Translators

- CPUs are very impressive but they are actually quite simple when it comes to processing. They can only process 1's and 0's (machine code). They therefore do not understand how to process high level programming code in the form in which we write it (e.g. Python).

High Level Language Translators

- Translators are programs that convert high level language commands, such as PRINT, IF and FOR, into a set of machine code commands such as 1011, 11001 and 11000011110, so that the CPU can process the data.

- There are 2 ways in which translators work:
  - Take the whole code and convert it into machine code before running it (known as compiling).
  - Take the code one instruction at a time, translate and run the instruction, before translating the next instruction (known as interpreting).

Compilers vs Interpreters

Compiler

- Converts the whole code into one file (often an .exe file).
- The file can then be run on any computer without the translator needing to be present.
- Can take a long time to compile source code as the translator will often have to convert the instructions into various sets of machine code as different CPUs will understand instructions with different machine code from one another.

Interpreter

- Converts the source code into machine code 1 line at a time.
- Program therefore runs very slowly.
- Main reason why an interpreter is used is at the testing / development stage.
- Programmers can quickly identify errors and fix them.
- The translator must be present on the computer for the program to be run.

Section 5: Questions

- Can computers understand high level programming code? Explain your answer. [2]

- Explain the similarity and differences between ‘Compilers’ and ‘Interpreters’. [3]

The End!

- You have reached the end, well done! (or if you just scrolled all the way to the bottom to see how many slides there are, there are 23, now get cracking!)

- As I mentioned, these topics are tough, do not worry if you struggle with them at this stage, this is to give you a taster of what to expect, and well… Computer Science is tough, but very rewarding!

We look forward to seeing you soon,

Mr Millington and Mrs Spooner