Tiverton High School Year 8 **Computing** **Autumn Term 1** Knowledge Organiser

Parts of a computer system | Key Construct 3: **Computer Systems**

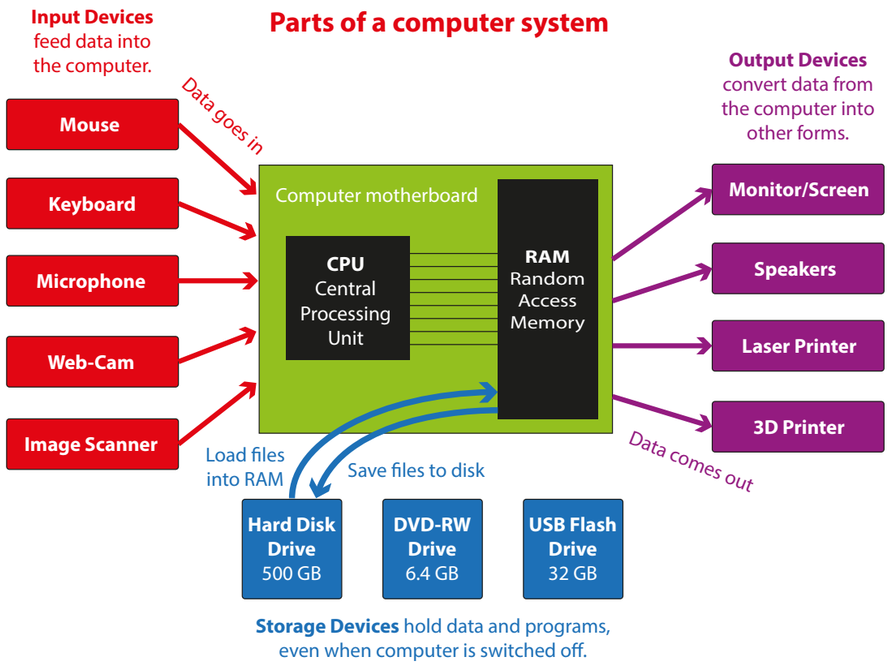
A **computer** is an **electronic device** that follows a **stored program of instructions**.
The **program** of instructions tell it **how** to process data and how to make things happen e.g. activate outputs.
A **computer system** is a **collection** of parts that **work together** to perform a task – comprised of **hardware** and **software**.

Input devices transfer data **into** the computer processor.
Examples: **QWERTY keyboard**, **mouse**, **microphone**, **web-cam**, **image scanner**, **accelerometer**, **fingerprint sensor**.

Output devices transfer data **out** of the computer for people to use.
Examples: **Screen/monitor**, **laser printer**, **audio speakers**, **3D printer**, **robot-arm**, **LED display**, **laser cutter**.

Storage devices store data for **long term** or **while the computer is switched off**.
Examples: **Hard-disk drive**, **solid-state drive**, **optical drive**, **USB Flash-drive**, **magnetic tape drive**.

Parts of a computer system



Computers in control | Key Construct 3: **Computer Systems**

An **embedded system** is a dedicated single-purpose computer that is **built into some other electronic device**.
The embedded computer **controls** the operation of that device.

Just like a general-purpose computer, the embedded system has a **processor**.
It has a small amount of **RAM** to hold data values it is using.
It may allow **simple inputs** using **buttons** or **dials**. It may **display simple outputs** on an **LCD screen** or **LED lights**.
It may also produce **simple audio outputs** and **sounds**.

Examples of embedded systems include **microwave ovens**, **burglar alarms**, **digital TV receiver boxes**, **scanner pens**, **drum machines**, **GPS sat-nav systems**, **traffic lights**, **elevators**.

Parts inside a computer | Key Construct 3: **Computer Systems**

Hardware means the **physical components, devices and circuitry** of the computer system.

A computer has a **processor** inside it. Another name for it is the **Central Processing Unit** (**CPU**).
The processor **executes** each instruction to carry out a program.

Processor speed is measured in **Hertz** (**Hz**)... cycles per second.

1 Hz (**Hertz**) = 1 clock cycle per second (very slow!)
1 MHz (**Mega-Hertz**) = 1 **million** clock cycles per second.
1 GHz (**Giga-Hertz**) = 1 **billion** clock cycles per second.

RAM stands for **Random Access Memory**. RAM is a kind of **memory storage** inside the computer.
We **"load"** programs and data values from secondary-storage into RAM, ready to use them.
RAM is used to hold the program of instructions that the CPU is running at the moment and the data it needs to use.
RAM is **volatile** - **all data is lost when the power is turned off**.

Creating web-pages using HTML | Key Construct 2: **Working with Software and Documents**

Web-pages are **viewed** using a program called a **web-browser**, such as **Internet Explorer**, **Google Chrome** or **Apple Safari**.
Most web-pages are created using a language called **HTML**, which stands for **Hyper-Text Markup Language**.
To **create** a web-page, you must write code in a **text-editor**, such as **Notepad**, **Notepad++** or **DreamWeaver**.
When you **save** a new web-page, the **filename** must always end in **.html** so web-browsers recognise it as a web-page.

You can add **special markers** in your text called **tags**. Each HTML tag tells your web-browser **how** to display something.
Tags are always enclosed in **< >** **angle-braces** e.g. `<TABLE>` means **start** making a **table**.
Many tags must be used as part of a **pair**. They control when to **start** doing something, and when to **stop** doing something.
Putting **/** in a tag means **stop** e.g. `</TABLE>` means **stop** making the table.

`<HTML>` `</HTML>` Starts and ends a new web-page.

There are **two main parts** to most web-pages: the **head** section and the **body** section.
The **head** **controls** certain things about the web-page.
The **body** contains the **content** that you can see in the web-browser window.

`<HEAD>` `</HEAD>` Starts and ends the **invisible** control settings section of the page.

`<BODY>` `</BODY>` Starts and ends the main **visible** part of your page, which will be displayed in the web-browser window.

`<TITLE>` `</TITLE>` Starts and ends the browser **window title**, which must be placed inside the **head** section.

`<BR>` **Line-break** - starts a brand new line of text in the page.

`<HR>` **Horizontal Rule** - draws a dividing line in-between sections of your page..

`<H1> </H1>` Starts and ends a **large headline/heading** e.g. `<H1>Welcome!</H1>`
`<H3> </H3>` Starts and ends a **smaller headline/sub-heading** e.g. `<H3>`How to contact us`</H3>`
`<B> </B>` Starts and ends some **bold** chunky text e.g. Bold words stand out because they look `<B>`chunky`</B>`
`<I> </I>` Starts and ends some *italicized* (sloping) text.

`<IMG src="smiley.jpg">` Inserts a **full-size image** (a **picture**) that has the filename "smiley.jpg".

`<IMG src="smiley.jpg" width=50 height=120>`
Inserts a picture,. You can set the **width** and the **height** to control how large the picture will appear in the web-browser.

`<A href="http://www.google.com">` click here `</A>` Makes text into a **hyper-link** that you can click on.

Tiverton High School Year 8 **Computing**
Autumn Term 2 Knowledge Organiser

Different kinds of software programs | Key Construct 2: **Working with Software and Documents**

An **operating system** is a program that makes your computer or your phone **easier to use**.

Examples of **operating systems** include:

- **Microsoft Windows** and **Ubuntu Linux** for **desktop PCs** and **laptops**;
- **Apple MacOS** for **Apple Mac computers** and **Macbook laptops**;
- **Google Android** for **Samsung phones** and **tablets**;
- **Apple iOS** for **Apple iPhones** and **iPad tablets**.

The operating system software **automatically** loads into your computer's memory as soon as you switch the computer on.
when the computer is loading the operating system, we say that the computer is "**booting up**" - it is getting ready for you to use.

Once the operating system has finished loading and it is running, the computer is ready to use. You can now load any program that you would like to use.

Without an operating system, most computers would be **too difficult to use** because they are **very** complicated machines.

Software applications are **general purpose programs**. They can be applied to solve many different kinds of problems.
Small application programs that you use on your **phone** or **tablet** are called **"apps"**.

Some examples of software applications include:

- **Word Processing** applications - to create text documents;
- **Spreadsheet** applications - to work with numbers, data, calculations, statistics, graphs and charts;
- **Databases** - to store and search through large amounts of information about people or things;
- **Presentation software** - to display facts, figures, pictures, charts and video clips on a large screen to an audience;
- **Graphics Packages**;- to create and edit photographs, illustrations and diagrams;
- **Audio and video editors** - to edit sounds, music, podcasts and video recordings.

Web-pages can be **displayed** or **viewed** in a program called a **web-browser**.
Examples of web-browsers include **Microsoft Edge**, **Apple Safari** and **Google Chrome**.

File-types and compression | Key Construct 5: **Data Representation**

A **file** is a **persistant** store of data that is held on a **secondary storage device** e.g. on a hard-disk drive.

Music and video files can contain a lot of data. Large files and streams of data can take a long time to transfer over the Internet.
If the file can be **compressed**, either by **reorganising** or **reducing the amount of data**, then it can be sent and received **faster**.

Compression re-organises a file of data and saves it as a new compressed file.
The compressed file usually has a **smaller file size** than the original.

It takes the computer **time** to compress the data – it's got to work out how to organise the data in a more **efficient** way.
Before you can use the data again, the computer needs to **de-compress** the file.
It must **re-organise the data** again into a form that can be used easily.

Text Documents

- **.txt** is an uncompressed **plain text document**. The text file contains only **unformatted text characters**.
- **.rtf** is an uncompressed **rich-text file**.
  The text file contains characters which can be **formatted** using **bold**, **italics**, **colour**, **font sizes** etc.
- **.pdf** is an Adobe **Portable Document Format** file.
  It can hold **rich-text**, **font definitions** and high-quality **vector diagrams**.
  Because the file contains the **font definitions** for each font face used it is **portable** - the document will look the same, regardless of the type of computer or phone being used.
  PDF files can also compress text and pictures to reduce the amount of data that they hold.

Images/Pictures

- **.bmp** is an uncompressed **bitmap image** format used widely by Microsoft Windows programs.
- **.tif** is an uncompressed **high-quality bitmap image** that can contain **millions of colours**.
  TIFF file sizes can be very large as they often contain so much uncompressed data.
- **.jpg** is a bitmap image that uses **compression**. Some of the finer details may be **lost** when compressing the image.
  JPEGs are used widely for **photographs** and can include **millions of colours**, making pictures very **realistic**.
- **.gif** is a compressed bitmap image that can only use up to **256 different colours**.
  This is only suitable for **simple graphics** and **animations**, or regions of **flat colour** that are all the same.
- **.png** is a **Portable Network Graphic**. This stores **high-quality graphics** using one or more **separate layers**.

Audio/Sound/Music

- **.wav** is an **uncompressed audio waveform**. These files are often very large, but result in **high-quality audio**.
- **.mp3** is an **audio file** that uses **compression**.
  The MP3 file is usually approximately 10 times smaller than their original. The sound quality can be quite low.

Video/Movies

- **.avi** is an **uncompressed video file** used widely by Microsoft Windows programs.
- **.mp4** is a **video file** that uses **compression**.

Programs

- **.exe** is an uncompressed **executable program file**.

Units of data storage | Key Construct 5: **Data Representation**

"bit" means **"binary digit"**.
A bit is the smallest amount of data that a computer can store.
A bit can either be a **0** or a **1** value. It uses an **ON** or **OFF** voltage in a circuit.

8-bit binary means a pattern of **8 binary-digits**.

8-bits allow **256 possible combinations** between **00000000** and **11111111**.

This is why 8 bits can represent between **0** and **255** in base ten.

1 byte	=	**8 bits** (an ASCII character takes 1 byte)
1 kilobyte	=	**1000 bytes**
1 megabyte	=	**1000 kilobytes** (or 1000 x 1000 bytes)
1 gigabyte	=	**1000 megabytes** (or 1000 x 1000 x 1000 bytes)
1 terabyte	=	**1000 gigabytes** (or 1000 x 1000 x 1000 x 1000 bytes)

Number bases

Binary means base-2
Computers use binary to store all data.

Denary means base-10
People usually use denary in everyday life.

Converting Binary (base two) to Denary (base ten)

128	64	32	16	8	4	2	1
0	**1**	**0**	**0**	**1**	**0**	**1**	**0**

(**1** x 64) + (**1** x 8) + (**1** x 2) = **74** in base ten

Tiverton High School Year 8 **Computing** **Spring Term** Knowledge Organiser

Types of Network | Key Construct 4: **Networks and Communication**

A **network** is a **collection** of two or more computer devices that are **connected** together.
Networked devices can share resources, programs and data e.g. **printers**, **databases of information**, collections of **documents**.

A **Personal Area Network (PAN)** is a **very small network**, connecting **only 2 or 3 devices** that are **used by a single person**.
You ake a Personal Area Network when you connect your mobile phone to Bluetooth headphones or to a smart-watch.

A **Local Area Network (LAN)** covers **one single site** (which may include a small number of buildings that are close together).
The distances between devices in a LAN are usually quite small.
Devices are often very close together, in the same room or building.
To make a Local Area Network, you can connect computer devices together with a **switch** or with a **wireless access point**.

A **Wide Area Network (WAN)** covers a **much larger physical area** than a LAN.
A Wide Area Network can **link devices together over long distances**. Devices may be many miles apart on different physical sites.
A WAN can cover a whole city, a county, a country or many countries.
Supermarkets, **banks**, **County Councils**, the **police** and the **NHS** all have their own Wide Area Networks.

The **Internet** is a massive collection of networks **all over the world**. The networks are **connected together** so that data can be sent from one network to another, wherever they are.

Transmission Media | Key Construct 4: **Networks and Communication**

Connections can be made between devices using different methods:

Copper cables

Data is transmitted over copper cables using **electricity**.
Copper cables allow relatively fast transmission but can be affected by other cables nearby, by **electrical interference** or **lightning strikes**.
Copper cables can make it very easy to connect new devices and are very cheap, but they can only be used over relatively short distances as the transmitted data signals weaken with distance e.g. up to 100m.

Fibre-optic cables

High-speed, high-capacity bundles of **glass fibre** that carry data as **light**, rather than electricity.
Fibre-optic cables are expensive and technical to install, but can be used reliably over very long distances, such as connecting different countries together under the sea.
Fibre-optic allows **extremely fast data transmission** and many people can share the use of the same cable due to its massive data capacity.

Wi-Fi links

Data is send and recieved using **radio signals** rather than using any cables.
Wi-Fi is ideal for situations where it is difficult to install cables, such as in old buildings.
Many devices can be configured to use a single **Wireless Access Point** device.
Wi-Fi transmission is **slower** than using copper cables or fibre-optic.

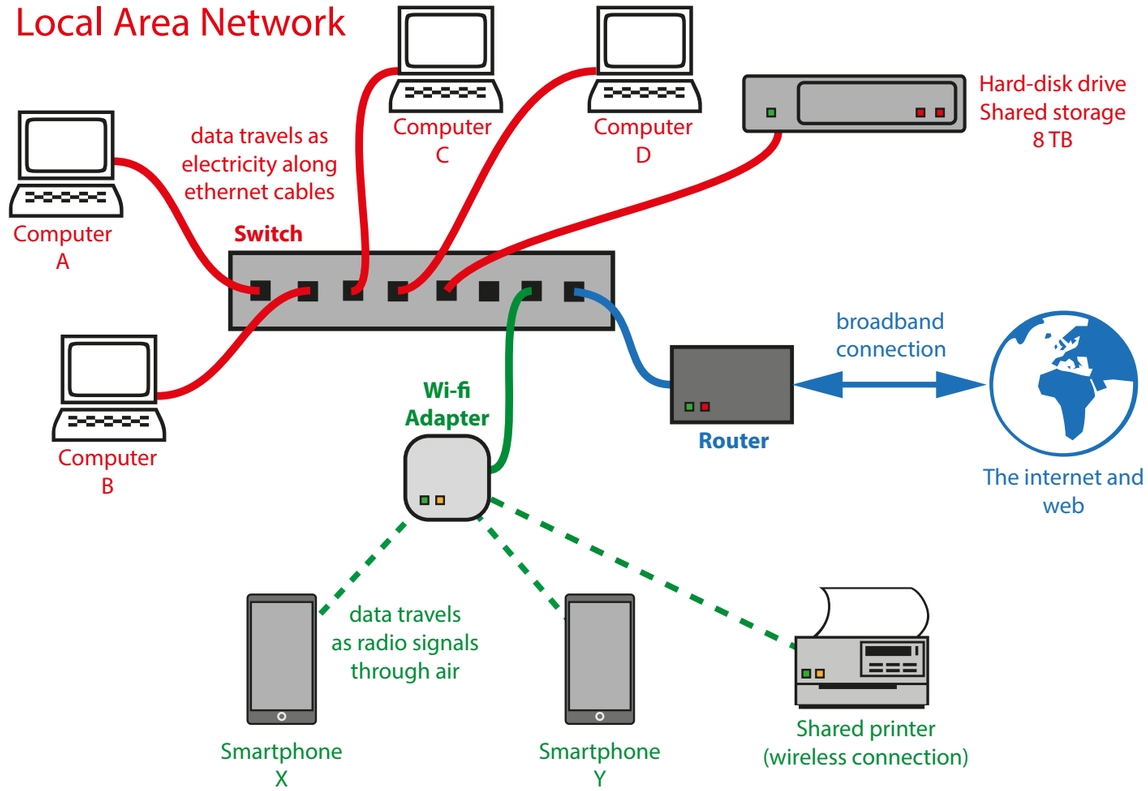
Micro-wave links

These are slow radio links that allow data to travel over **long distances** e.g. to and from satellites in space.
Micro-waves are useful where it would be very difficult to install a cable e.g. for a moving ship at sea.
Micro-wave reciever dishes usually need to be lined up precisely to focus and recieve data signals.
Micro-wave transmitters and recievers are relatively expensive, but they can be used **anywhere on Earth**.

Bluetooth

Bluetooth is a particular kind of **very high-frequency radio link** that allows data to be exchanged between personal devices. It is relatively slow, but can be used for **wireless keyboards** and **mice**, **headphones** and **smartphone-to-computer file transfers**. Bluetooth devices are often **paired** together, exchanging setup information to make communication more secure and to stop unauthorised access.

Local Area Network



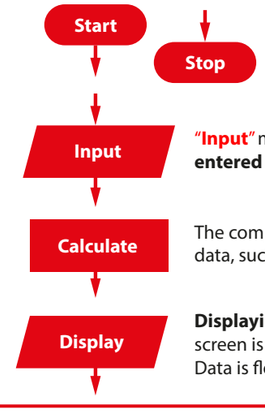
Problem solving and computational thinking | Key Construct 6: **Problem Solving and Programming**

Decomposition means **breaking a problem down into smaller parts**, which are easier to solve.

Abstraction means **choosing only the most important details that are relevant to solving the problem**, while **ignoring** other details.

A **program** is a sequence of instructions that the computer will carry out (execute).
An **algorithm** is a precise set of written steps that describe exactly how to solve a problem.
A **flowchart** is a diagram that shows how an algorithm works.

Flowchart symbols | Key Construct 6: **Problem Solving and Programming**



A flowchart always begins with a **"terminator"** shape to mark the **beginning** or **end** of the flowchart.

"Input" means **data is entered into the computer**.

The computer can **process** some data, such as use it in a **calculation**.

Displaying things on the screen is a kind of **"output"**.
Data is flowing **out** of the computer.

A **decision** diamond often has different routes coming out of it, such as "**Yes**" and "**No**"

Tiverton High School Year 8 **Computing** **Summer Term** Knowledge Organiser

Important programming ideas | Key Construct 6: **Problem Solving and Programming**

You can **create** software by writing new programs. You tell the computer what to do, step-by-step, giving it instructions that it will follow.

When you have finished making your program of instructions, you can **run** through them, asking the computer to carry the instructions out one-at-a-time in order. This is called **executing** a program.

You write the program instructions using a **programming language**. You can't just write your program instructions using ordinary english language because many sentences in the english language are too complex for a computer to break down.

There are lots of different programming languages that you can use to make a new program.
Each one has different advantages but some are more difficult to learn than others.

Small BASIC and **Python** are two programming languages that are quite easy to learn for beginners.

To write your program instructions for a new program, you must type them carefully into a **text editor**.

If you make a **mistake**, an instruction may not make sense to a computer. This is called a **syntax error**.
When you try to run your program, the computer can tell you if it finds a syntax error.
The computer may suggest which line in your program needs to be fixed. You can then look for mistakes in your typing.

Programming techniques you can use when writing programs | Key Construct 6: **Problem Solving and Programming**

Input means gathering some data from the keyboard or other input device and storing it in a **variable**.

Output often means **displaying** something on the **screen**.

You can display **words**, **numbers**, or the value of **variables** that are stored inside the computer's memory.
This is called **text output**.

Many programming languages can also be used to **draw lines** and **shapes** on the screen.
This is called **graphical output**.

A **sequence** is a group of program statements that are executed in the **correct order**, one after the other.

A **variable** is a **named value** that can **change** while your program is running e.g. **score**

Assignment means giving a **value** to a variable.

Iteration means repeatedly executing parts of the program **again and again**. This is sometimes called a **loop**.

Selection means making a **decision** to select which part of the program code should be executed.

When you tell the computer to calculate something, you need to use the correct symbol.
Programming languages sometimes use different symbols to those you usualy use in a Maths lesson.

Performing arithmetic and calculating

+ Addition **- Subtraction** *** Multiplication** **/ Division**

Symbols to help the computer make comparisons between things

< less than **> greater than** **<= less than** or **equal to** **>= greater than** or **equal to**

Writing simple programs in Small BASIC | Key Construct 6: **Problem Solving and Programming**

Assignment - storing a value in a variable inside the memory of the computer:

```
score = 10
password = "Cu5tArd"
```

Text values must always be enclosed between " **speech marks**), this shows the computer where the text begins and ends, even if the text contains spaces.

Input - **gathering** a new **number** or **text value** andstoring them using **variables**:

```
mynum = TextWindow.ReadNumber()
mytext = TextWindow.Read()
```

Output - **displaying** text messages or the value of a variable on the screen:

```
TextWindow.WriteLine("GAME OVER!")
TextWindow.WriteLine(mynum)
```

Iteration - **repeatedly** executing something a certain number of times:

```
For number = 0 To 100 Step 10
      TextWindow.WriteLine(number)
EndFor
```

Iteration - **repeatedly** executing something until something special happens:

```
While time < 60
      TextWindow.WriteLine(time)
      time = time - 5
EndWhile
```

Selection means **making a decision** to **select which part of the program code should be executed**:

```
If lives > 0 Then
      TextWindow.WriteLine("Lost a life!")
Else
      TextWindow.WriteLine("Game Over")
EndIf
```

A **sub-routine** is a **part** of your program. You can give the sub-routine a **name** to describe what it does.

```
Sub DrawOneSquare
      For side = 1 To 4
            Turtle.Draw(100)
            Turtle.Turn(90)
      EndFor
EndSub
```

After you have made a **sub-routine**, you can **activate** it whenever you like from anywhere in your program code.
Use the name of the sub-routine to activate it. This is also known as **"calling"** a sub-routine (you **call** it into action).

```
For shape = 1 To 10
      DrawOneSquare()
      Turtle.Turn(36)
EndFor
```