

Tiverton High School Year 7 Computing Autumn Term Knowledge Organiser

Hardware and software

Key Construct 3: Computer Systems

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

Software means the **programs** that it uses.

A **program** is a set of **instructions** that tells the computer **what to do**.

Without software, the hardware would be useless, it would not have any instructions to follow.

Operating systems, device drivers and **utilities** are examples of **systems software**.

Without an operating system, a modern computer would be too difficult to use.

General purpose **software applications** include **Word Processing, Spreadsheets, Graphics Packages**.

Devices and components

Key Construct 3: Computer Systems

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

Output Devices transfer **out** of the computer for people to use.
Examples: **Screen/monitor, speakers, printer, 3D printer, LED**.

Storage Devices store files even while the computer is off.
Examples: **Magnetic hard-disk drive, solid-state drive, USB flash-drive, CD-ROM drive, DVD-ROM drive**.

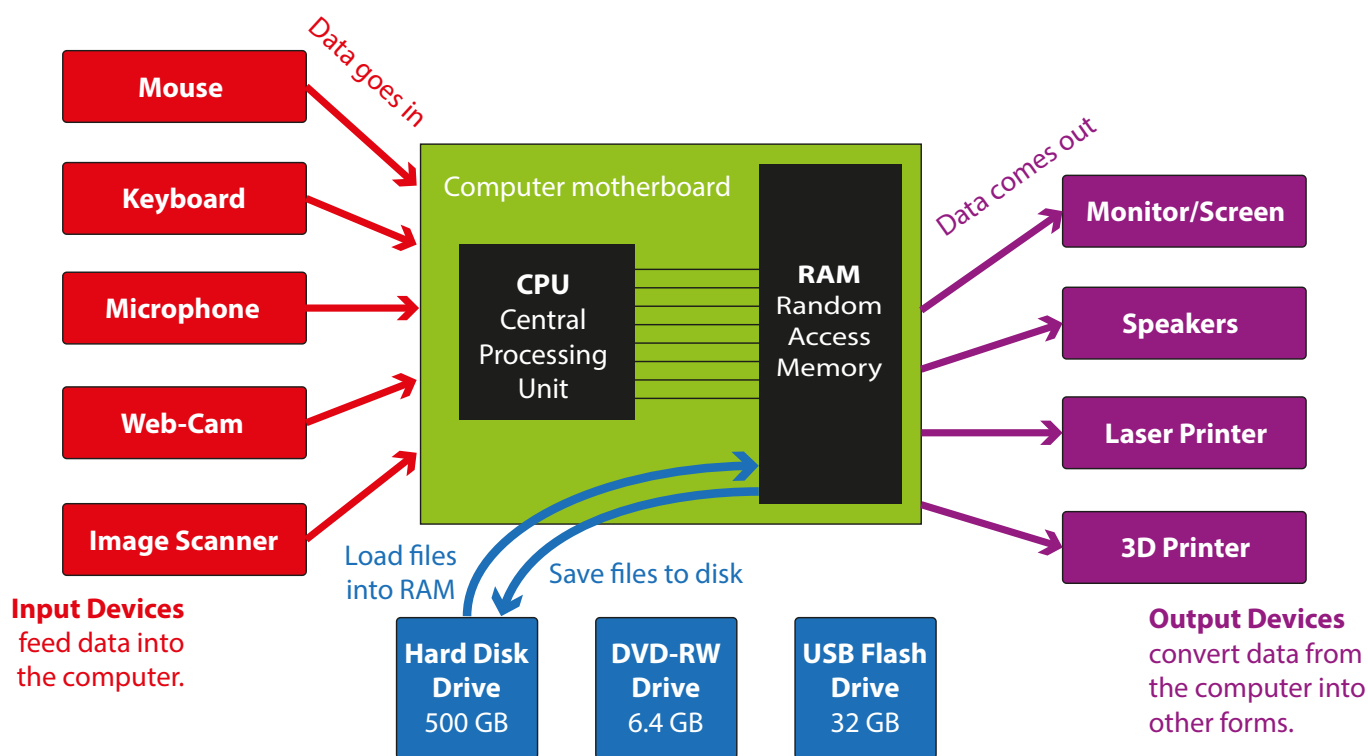
A computer has a **processor** inside it (called a **CPU**).
The processor **executes** each instruction to carry out a program.

RAM stands for **Random Access Memory**

The RAM inside a computer holds the program of instructions that the CPU needs to carry out.

RAM is **volatile...** all data is lost when the power is turned off.

Parts of a computer system



Storage Devices hold data and programs, even when computer is switched off.

Working with text | Key Construct 1: Working with Software and Documents

We use **Microsoft Word** to prepare most written documents. Microsoft Word is a **word-processor**.

It also lets you add **pictures, tables, page numbers** and other things to your document.

Always make it clear what your document is about. Use a large, bold **heading**.

Divide your writing into different **sections**. Label each new section with a smaller **sub-heading**.

Choose a **font-face** that is easy to read, such as **Calibri** or **Arial**.

Use a consistent **font size** for the main text in your document. Try **10 pt** or **11 pt**.

Use **bold** or *italics* to make important parts of your text stand out and gain attention from the reader.

You can organize more complicated information using a **table**.

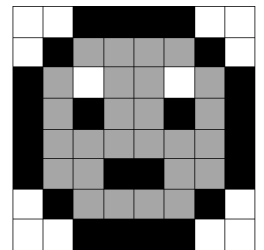
Working with pictures | Key Construct 1: Working with Software and Documents

Bitmap images are **pictures** that are made up of **pixels** (picture elements).

A **pixel** is a small coloured dot in a picture.

All of the pixels are arranged in a grid, a little bit like a mosaic.

The **colour** of each pixel is stored in the memory of the computer using binary digits... **1s** and **0s**.



Working with numbers | Key Construct 1: Working with Software and Documents

We use **Microsoft Excel** to perform **calculations**, produce **statistics** and plot **graphs** and **charts** from data values. Microsoft Excel is a **spreadsheet program**.

It can **multiply, divide, add, subtract** and work out **averages**. It can also make **decisions** about data.

A spreadsheet uses a grid of **cells**. A cell is like a "**box**".

Each cell can hold one **data value** - which is often either a **number** or a short piece of **text**.

A whole **vertical** line of cells is called a **column**.

A whole **horizontal** line of cells is called a **row**.

Column headers are labelled with **letters**.

Row headers are labelled with **numbers**.

Using the column letter and the row number you can find one cell. This is called a **cell reference**.

Cell **C5** is in **column C**, **row number 5**.

Cell **G23** is in **column G**, **row number 23**.

A spreadsheet can **calculate** things for you. Instead of typing in a data value into a cell, you can type in a **formula**.

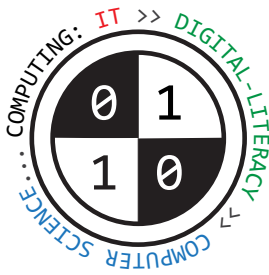
A formula always begins with the **=** sign.

This sign tells the spreadsheet to **work something out for you**.

So the formula **=B5*2.2** would find what is in cell **B5** and then **multiply** it by **2.2** for you.

Remember: ***** means "multiply" **/** means "divide"

	A	B	C
1	Conversions		
2		Mass (kg)	Mass (lbs)
3		2.5	5.5
4		5	11
5		10	=B5*2.2
6			



Tiverton High School Year 7 Computing Spring/Summer Terms Knowledge Organiser

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

Web-pages can be **displayed** or **viewed** in a program called a **web-browser**.



Web-pages can be **created** using a special language called **HTML** (Hyper Text Markup Language).

When creating a web-page, a person adds special codes called "HTML **tags**" into their document. The tags tell the web-browser exactly **how** to display parts of the document.

A web-page is made up of 2 separate parts: the **HEAD** and the **BODY**.

The **HEAD** section contains **important settings** about the web-page that you cannot actually see in the main browser window.

The **BODY** part contains **all the information that you will be able to see when the web-browser loads the web-page**.

Some tags can also contain extra useful information for the browser to use. These details are placed **inside** the tag. They are called **attributes**. A good example is the **IMG** tag. When typing an **IMG** tag into your web-page, you can add extra information to set the **width** and the **height** of the picture so the web-browser displays it correctly.

HTML tags for creating web-pages | Key Construct 2: Working with Software and Documents

<HTML> </HTML> makes a web-page that can be displayed in a web-browser program.

<HEAD> </HEAD> makes the **head** section of the web-page. This holds important **settings** for the web-page.

<BODY> </BODY> makes the **body** of the web-page. Anything inside the body section will be **displayed** in the web-browser.

<TITLE> </TITLE> must be used **inside** the head section. This sets the **title** (or **name**) of the browser window.

<H1>Exclusive!</H1> makes a large **heading/headline**.

**** inserts a **picture** into the page, setting out how large it should be.

Click here creates a **hyperlink** so a person can **jump** to another web-page.

Here is an example of a very simple web-page made using HTML:

```
<HTML>
  <HEAD>
    <TITLE>Web-Page!</TITLE>
  </HEAD>
  <BODY>
    <H1>Look!</H1>
    This is a simple web-page that can be displayed in a browser.
  </BODY>
</HTML>
```

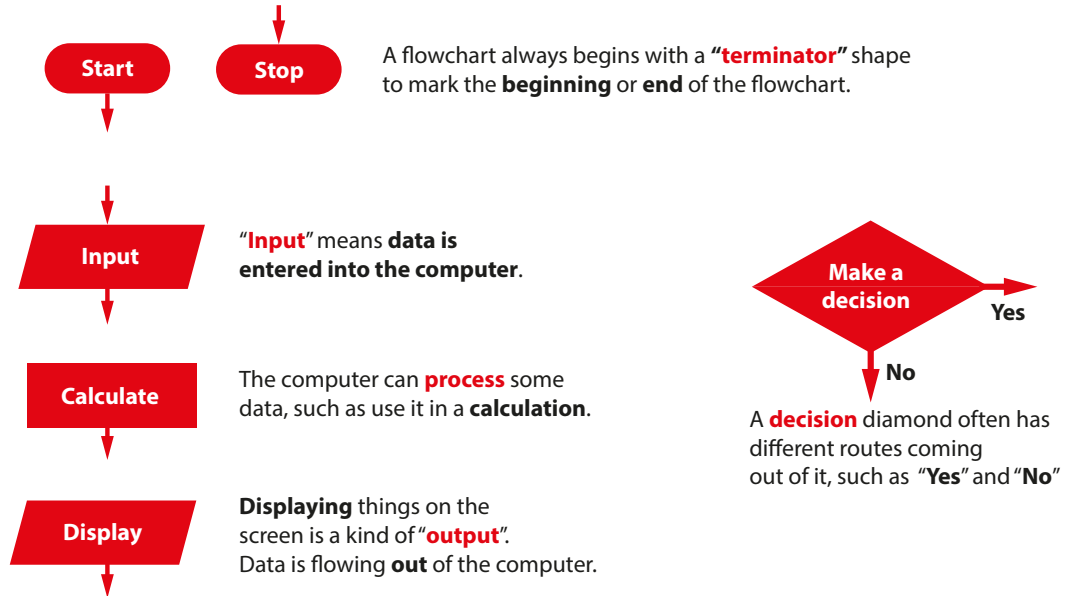
Planning solutions to problems | Key Construct 6: Problem Solving and Programming

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



Creating and testing computer programs | Key Construct 6: Problem Solving and Programming

You can create software by writing new programs. You write the program instructions using a **programming language**.

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

```
INPUT width
```

Output means displaying something on the screen:

```
PRINT "Your final score is"  
PRINT score
```

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

```
x = 3          password = "Cu5tArd"
```

Iteration means repeatedly executing parts of the program again and again (looping):

```
FOR time = 1 TO 10      WHILE time < 60
```

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

```
IF lives > 0 THEN  
    PRINT "Lost a life"  
ELSE  
    PRINT "Game Over"  
ENDIF
```

Arithmetic operators

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

Relational Operator Symbols when making comparisons

<	less than	>	greater than
<=	less than or equal to	>=	greater than or equal to
==	is the same as	!=	not the same as