Tiverton High School Year 7 **Computing**
Autumn Term Knowledge Organiser

How we use passwords | Key Construct 1: **Impact of Digital Technlogy**

Choose a password that **you** will be able to **remember**.

Don't make the password too short. Short passwords may be easier to guess.

Use a **mixture** of **different** kinds of **characters**:

capital letters **ABCDEFGHIJKLMNOPQRSTUVWXYZ**

lower-case letters **abcdefghijklmnopqrstuvwxyz**

digits **0123456789**

other symbols **$£@!**

Examples: `Iquito$96` `br@mbleD0g` `SFpyram1d` `iqu1t0$96`

Remember...

Don't tell anyone your password.
Don't write it down because people could see it.
Don't let people watch you when you type it in.
Change your password if you think someone knows it.

Working with text | Key Construct 2: **Working with Software and Documents**

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

A word-processor lets you add **pictures**, **tables**, **page numbers** and other things to your **writing**.

Always make it obvious what your document is about. Start your document with a large, bold **heading**.

Break 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**.

You can add special **scientific symbols** or characters that are used in **different languages**:

á ç é ö ∞ Δ π √ α β γ

You can **raise** characters **above** the rest of your writing. This is called **super-script**:

cm^3 $y = x^2 + 5$

You can also **lower** characters **below** the rest of your writing. This is called **sub-script**:

O_2 H_2O CO_2 H_2SO_4

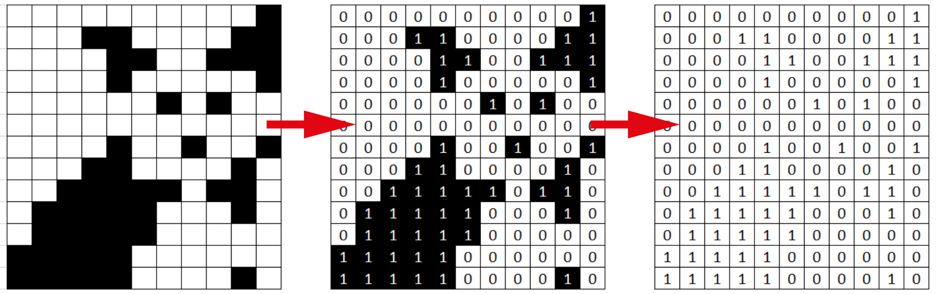
Working with pictures | Key Construct 2: **Working with Software and Documents**

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 a apattern of **binary digits**... **1**s and **0**s.



0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	0	0	1	1
0	0	0	0	1	1	0	0	1	1	1
0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1	0	0	1
0	0	0	1	1	0	0	0	0	1	0
0	0	1	1	1	1	1	0	1	1	0
0	1	1	1	1	1	0	0	0	1	0
0	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	1	0

0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	0	0	1	1
0	0	0	0	1	1	0	0	1	1	1
0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1	0	0	1
0	0	0	1	1	0	0	0	0	1	0
0	0	1	1	1	1	1	0	1	1	0
0	1	1	1	1	1	0	0	0	1	0
0	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	1	0

Working with numbers and data values | Key Construct 2: **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**.

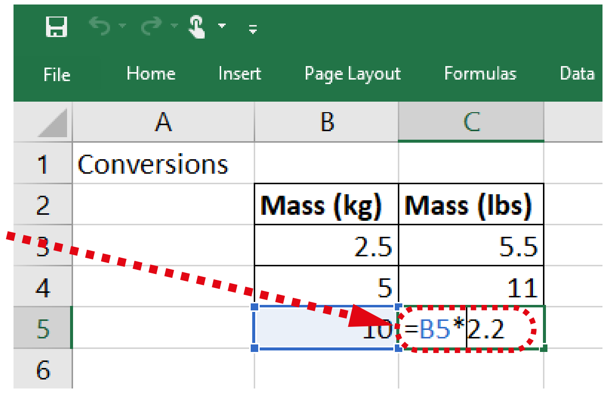
Using the column letter and the row number you can find one cell. This is called a **cell reference**.
Cell **C5** is in **column C**, **row number 5**.
Cell **G23** is in **column G**, **row number 23**.

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			

`=SUM(B3:B5)` would work out the **sum total** of all the numbers in the **block** of cells from **B3**, to **B5**.

`=AVERAGE(B3:B5)` would work out the **mean average** of all the numbers in the **block** of cells from **B3**, to **B5**.

`=MIN(B3:B5)` would find the **minimum value** (the **smallest** value) in the block of cells from **B3** to **B5**.

`=MAX(B3:B5)` would find the **maximum value** (the **largest** value) in the block of cells from **B3** to **B5**.

Tiverton High School Year 7 **Computing**
Spring Term Knowledge Organiser

Hardware and software | Key Construct 3: **Computer Systems**

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

Software means the **programs** that a computer uses. Software is not physical.

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

Without any software programs, the computer hardware would be **useless**.
The computer would not have any instructions to follow, so none of the circuits would activate.
An empty computer will not do anything when you switch on the power. It **needs** to have some software to tell it **what to do**.

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 computers are **very** complicated machines.

Software applications are general purpose programs. They can be applied to solve many different kinds of problems.
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;

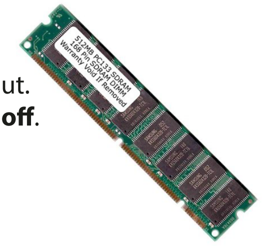
Graphics Packages;- to create and edit photographs, illustrations and diagrams;

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**.

Devices and components | Key Construct 3: **Computer Systems**

A computer must always have special circuitry inside it called the **processor**.
The processor is also called the **Central Processing Unit** or **CPU**.
The processor **executes** or **carries out** each instruction in a **program** to make things happen.



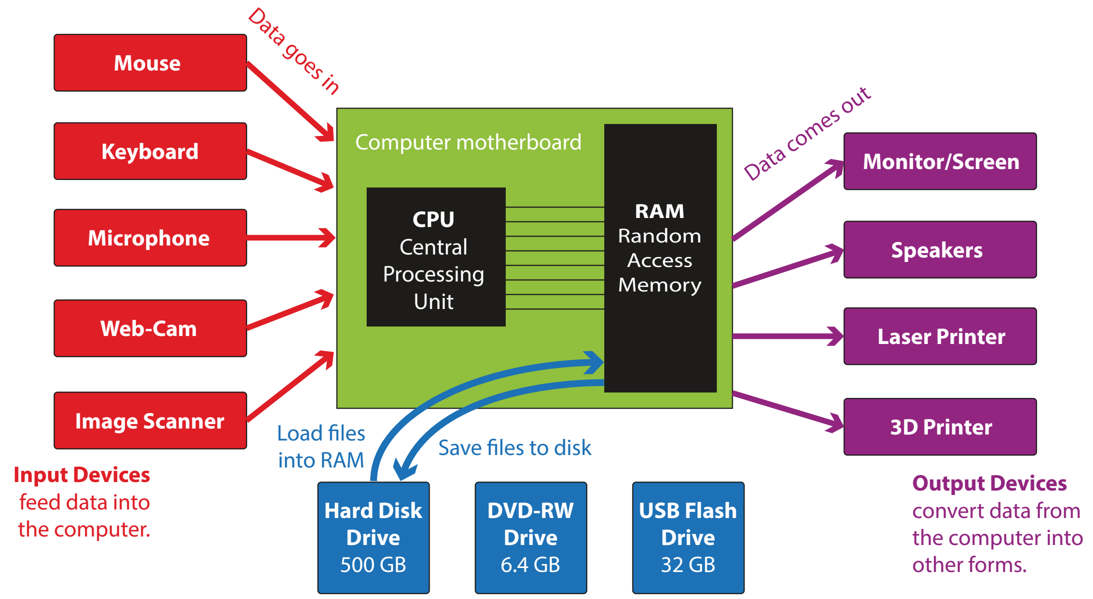
RAM stands for **Random Access Memory**
The RAM inside a computer holds the program of instructions that the CPU needs to carry out.
RAM is **volatile… all data that was stored in RAM will be lost when the power is turned off**.

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

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

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**.

Hardware that makes up a computer system



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

Tiverton High School Year 7 **Computing** **Summer Term** Knowledge Organiser

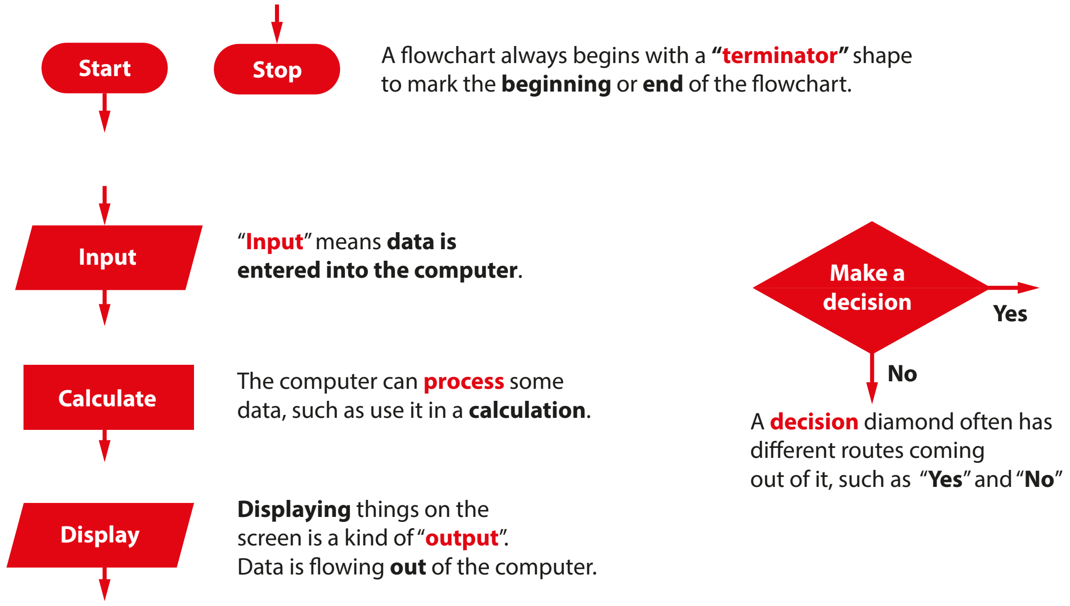
Planning solutions to problems | Key Construct 6: **Problem Solving and Programming**

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



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**"

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
```