Tiverton High School Year 9 **Computing** **Autumn Term** Knowledge Organiser **Part 1** **Number Bases and Data Representation**

Number bases and units of storage | Key Construct 5: **Data Representation**

Binary means base-2

Denary means base-10

Hexadecimal means base-16

Humans traditionally use **denary (base 10)** when dealing with numbers.
Computers always use **binary (base 2)** to store and process digital data.

Electronic computers contain millions of tiny **transistor** components.
A transistor behaves like a **switch**, that can only be turned **on** or **off**.
Because binary only uses two possible digits, these closely match the on/off states of the transistors that computers are made of.
The **on** or **off states** of transistors can be used to represent the two different number symbols that binary uses:

off means **0**
on means **1**

A **bit** is the **smallest** amount that a computer can store - one **binary digit**.

8-bit binary means a **pattern** of exactly **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)

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

Converting Hexadecimal (base sixteen) to Denary (base ten)

Hexadecimal is a more **compact** and **convenient** way to represent **large** numbers than binary.

Large numbers can be represented using **fewer** hexadecimal digits.

Hexadecimal numbers can only uses the symbols **0123456789ABCDEF**

A means **10**
B means **11**
C means **12**
D means **13**
E means **14**
F means **15**

16	1
2	**D**

2 groups of 16, plus **D** units.
(**2** x 16) + (**D** x 1)
(**2** x 16) + (**13** x 1)
32 + 13
45 in base 10.

How computers store text | Key Construct 5: **Data Representation**

A **character** is a **symbol** that can be represented and stored by the computer system.
The full collection of ALL of the characters that a computer can represent/store is called a **character set**.
Each character symbol is represented using a **special number** called a **character code**.

ASCII is the **American Standard Code for Information Interchange**. It can be used for writing in the English language.
Plain **ASCII text** is often stored using **7 bits** per character.

A better version of ASCII is **Extended ASCII**. This can be used for writing in **English**, **French**, **German**, **Spanish** or **Italian**.
Extended ASCII allows more characters than original ASCII, but uses **8 bits (1 byte)** to store each different character code.

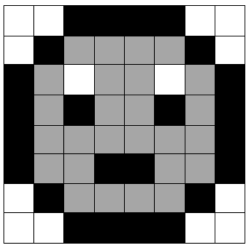
Lächeln **bonjour à tous** **¡Rápidamente!**

Unicode is a better character set. It can represent **any** language in the world, including Russian and Chinese, not just English.
Unicode uses up to **32 bits (4 bytes)** to store each character code.
Emoji pictures are character symbols from the Unicode character set. ASCII and Extended-ASCII do not contain any emojis.

中文 **р у с с к и й** **日本語** ☺

How computers store images | Key Construct 5: **Data Representation**

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... **1**s and **0**s.

The **bit-depth** of an image means **how many binary digits are used to store each pixel**.

A **1-bit image** uses exactly **1 bit to store each pixel** in the picture. This allows **2 possible colours**.
A **2-bit image** uses exactly **2 bits to store each pixel** in the picture. This allows **4 possible colours**.
An **8-bit image** uses exactly **8 bits to store each pixel** in the picture. This allows **256 possible colours**.

Photoshop uses **24-bit images**. It uses **24 bits to store each pixel**. This allows **16,777,216 possible colours** for **realistic** pictures.

Resolution means the **density** of the pixels in an image: **how many pixels will fit into a certain area**.
The **resolution** of an image defines how large the individual pixels are drawn.
The higher the resolution, the more life-like the image/better quality, but the more data will be included in the bitmap file.

Most computer **screens** use **72 dots per inch** - large pixels.
Many **printers** use **150 dots per inch** or **300 dots per inch** - the smaller pixels produce a more **detailed** picture on paper.

The **colour-model** used by a program controls **how colours are mixed together** to make pictures.

Most computer programs use the **RGB (Red-Green-Blue) colour model** to display images on the screen.
The colour of any pixel can be made by mixing **red light**, **green light** and **blue light** togther in varying amounts.

Many printers use the **CMYK model**. They combine the colours **Cyan**, **Magenta**, **Yellow** and **Black** in different amounts.

Many image files contain **extra data**, as well as the pixel data. The extra data is called **meta-data**.

Meta-data can be used by programs to **reconstruct** and display images from a file of binary data:
It includes the **width**, the **height**, the **resolution** and the **bit-depth** of the image.

Extra meta-data can also be included in an image file, such as the **file format**, the **date/time it was created**, **who owns the copyright** and the **GPS coordinates of where a photo was taken**.

How computer store audio (sounds and music) | Key Construct 5: **Data Representation**

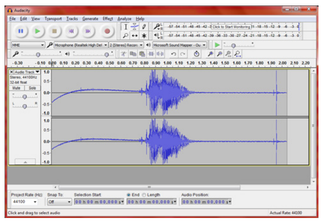
To represent audio/sound inside a computer, soundwaves are converted to digital data.

First of all, a sound wave must be captured by a microphone as electrical signals.
The **height of the sound wave** can then be **measured** at regular intervals. We call each measurement a **sample**.
The **number of sample measurements that are generated each second** is called the **sample-rate**. This is **measured** in **Hertz (Hz)**.
Each sample/measurement is stored in the computer using a binary number.
The **number of binary digits used in each sample** is called the **sample-size**.

A higher sample rate and sample size, leads to a larger audio-file, but a better-quality recording.
A realistic audio file will need to use thousands of samples a second. Common sample rates include **22050 Hz** or **44100 Hz**.

To reconstruct a sound from binary data, a audio file needs to contain extra meta-data that describes how the binary data is structured and how to play it back:

- **Duration** of the sound (how many **seconds** the recording lasts).
- **Sample-Rate** (**how many samples were used each second** e.g. **8000 Hz**).
- **Sample-Size** (**how many bits each sample contains** e.g. **32 bits**).
- **Channels** (**how many speakers are needed** e.g. **1 for mono**, **2 for stereo**).
- **Date & time** that the audio file was **created** or **last changed**.
- **Author**, **genre** or **copyright information** about **who** created the recording.



Tiverton High School Year 9 **Computing** **Spring Term** Knowledge Organiser

Decomposing problems and developing solutions | Key Construct 6: **Problem Solving and Programming**

A **program** is a **sequence of instructions** that the computer will carry out (**execute**).
Most programs need to be **planned** out very carefully in advance to make sure they don't crash or do the wrong thing.

Decomposition means **breaking a problem down into smaller parts**, until each part is easy enough to understand and solve.

Abstraction means **choosing only the most important details that are relevant to solving the problem**, while **ignoring** other details. When working out how to solve a problem, abstraction helps you avoid getting bogged down in too much detail.

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.

You can **plan** out the steps of a new program using **pseudo-code**... "false" code.
Pseudo-code is **not** a real **programming language** - you can't type pseudo-code into a computer and then run it.

The point of pseudo-code is that it lets you write out the **precise** steps that will solve a problem.
It helps people make sure they really **understand** the problem they are trying to solve **before** writing real program code.

Once you have written out a pseudo-code solution and checked it is correct, you are **much** less likely to build errors into your real program code.

You can then use your pseudo-code solution as a guide to help you to develop your real computer program using a programming language, such as **Python**, **BASIC**, **C**, **C++**, **C#** or **Java**.

Writing solutions to problems using pseudo-code | Key Construct 6: **Problem Solving and Programming**

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

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
PRINT "You have", lives, " left"
```

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 = "orR1bLe"
```

Iteration means **repeatedly** executing parts of the program again and again in a **loop**:

```
FOR time = 1 TO 10

WHILE time < 60
```

Selection means making a **decision**.
Your program can select which part of the program code should be executed next:

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

Developing computer programs using Python | Key Construct 6: **Problem Solving and Programming**

Python is a **high-level programming language**. It can be used by beginners to create computer programs.

Many people use **IDLE** to create Python code. This is an **Integrated Development Environment**. It contains a **text-editor** for writing Python code and other tools that are helpful to programmers.

When you **save** a new Python program, the filename needs to end in **.py** so the computer knows it can be **executed** using Python.

If you make a mistake or type an error in your program, the code may not make sense when Python tries to execute it.
Python will stop running your program and try to show you **where** the error is in your code so you can fix it.
This is called a **syntax error**.

A **comment** is a line of text in your program code that the computer **will not execute**. It will be used by the computer when running a Python program. It is used as a **reminder** or as an **explanation** to someone about how your code works.
To make text into a comment, type in the the **#** symbol at the start of the line of text.

```
# Main menu starts here
```

Your programs can work with different **kinds** of data values. We call these **data-types**.

Integer	a **whole number**	e.g. `32`, `-7`, `0`
Real	a number that can contain a **decimal point**, these are called "float" values in Python	e.g. `3.14`
Boolean	a value that can only hold one of two possible states, either `True` or `False`	
Character	a **SINGLE character symbol**.	e.g. one **letter**, one **digit**, a **punctuation mark**. `'?'`
String	a **sequence** of **characters**.	e.g. `'This cheese smells.'`

Most Python programs use **variables**. A variable is a named value that can be **changed** during the execution of a program.
When we set the value for a variable, we call this **assignment** (we are assigning a value).
Use the **= assignment operator** to **set a value** e.g. `password = '5ecRet123'`

A **print** statement **displays a line of text on the screen**. This is an **output**.
Text enclosed by `' '` quotes will literally be displayed. Anything without quotes will display the value of a variable.

```
print( 'Hello' )
print( password )
print( ‘You have ’, lives, ‘ lives left.’ )
```

An **input** statement allows people to type useful data **into** a program once it is running.
Whenever you type a data value into the computer, the value must be **stored** in a **variable** (so the computer does not lose it).

```
surname = input( 'Type in your surname... ' )
```

A program can make **decisions** while it is running to **choose** what should happen next. This is called **selection**.
It can **decide** whether or not something should happen, depending on whether a **condition** is found to be **True** or **False**.

```
if guesses <= 5 :
      print( 'Try again.' )
```

```
if guesses <= 5 :
      print( 'Try again.' )
else :
      print( 'No guesses left.' )
```

Iteration means to carry out instructions **more than once**.

You can carry them out a certain, definite number of times using a `for` loop.
A `for` loop always **counts** how many times something has happened.

```
for num in range( 1, 6 )  :
      print( num * 10 )
```

You can also carry out a sequence of instructions only while a certain **condition** holds **True**, using a `while` loop.

```
while keepGoing == ‘yes’ :
      keepGoing = input( ‘Do you want to keep running this program?’ )
```

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**

Tiverton High School Year 9 **Computing** **Summer Term** Knowledge Organiser

Storage devices | Key Construct 3: **Computer Systems**

Secondary storage devices are used for **long term storage** of data and instructions.

Programs and data are stored in **files**. Files are stored **even when the computer is switched off**.

We say that secondary storage is "**persistent**" or "**non-volatile**".

Magnetic Hard Disk Drive	A **high capacity** device that can often store as much as **8 TB** of data on one drive. Data is stored using tiny **magnetised** areas on a rapidly spinning metal disk. Magnetic hard-disks can be **damaged** accidentally by a **sudden impacts** or if it is dropped. Data can be **corrupted** or **erased** accidentally by **magnetic fields** from **speakers**, or **heat**.
Solid State Drive	An alternative to using a magnetic hard disk drive, but does not contain any moving parts. Data is stored using **tiny components** in **solid-state circuits** called **flash memory**. Solid-state drives are not affected by magnetic fields or extreme temperatures. They are very **lightweight** and **impact-proof**, making them ideal for use in laptops. They cannot hold quite as much data as magnetic disk drives and are **more expensive per GB**. Solid State Drives can sometimes start to **wear out** after data has been written to the same area a large number of times. Areas of the drive can then become **less reliable** for storing your data.
Flash Memory Card	A tiny, **portable memory card** that can be used to **transfer data between devices**. They can usually store between **64 GB** and **512 GB** of data, although some hold even more. They are used in **digital cameras** and **mobile phones**, but can be read by many laptops and PCs. They are **impact-proof** but can be damaged by **static-electricity** if not handled carefully.
USB Flash-Drive	A **removable storage device** that can be used to **transfer files from one computer to another.** They are similar to solid-state drives and flash memory cards. Their data is held in **flash memory**. Flash drives are often **encrypted** to prevent **breaches of sensitive data** if they are lost.
CD-ROM	A **removable optical disk** that stores data as tiny **pits**, burnt into the surface by a **laser beam**. A single CD-ROM can store as much as **900 MB** of data. Highly **portable**, making it ideal for **backing up files** or **transferring data** to other computers. Very **cheap to manufacture**, making them ideal to **distribute software utilities** and **audio**. CDs are **not very durable**. A **scratch** can make individual files or the whole disk **unreadable**.
DVD-ROM	A **removable optical disk**, similar to a CD-ROM, but with a much larger storage capacity. A single DVD-ROM can store enough compressed data for a **whole feature-length movie**. It can usually store at least **4.7 GB** of data, although some types of DVD can store much more. Because a DVD can hold more data than a CD, they are used as **installation disks** for software.
Blu-ray	An **removable optical disk** that can store enough data for **several hours of HD video**.

Laws that govern how we use computer technology | Key Construct 1: **Impact of Digital Technology**

The **Data Protection Act 2018** covers **how personal data may be used by companies and organisations**.
It describes **the type of data can be collected**, **how long data can be kept for** and the need to **keep data up to date/accurate**.
It sets out **restrictions on sending and using data**. It also **defines who is allowed to view or make use of data**.

The **Computer Misuse Act 1990 makes it illegal to use or to attempt to use computers to access computer systems without permission**. It also **make it illegal to access computer systems with intent to commit a criminal offence**, or to **alter data without permission** (e.g. through the use of viruses, physical deletion etc).

Compression and file-types | Key Construct 5: **Data Representation**

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.

Sometimes, parts of the original data are **removed** during compression. When the file is uncompressed again, some of the data will be **lost forever**. This is called **lossy compression**. The data that was removed can **never** be recovered again.

When compressing **executable programs** and **text documents** we need to use **loss-less compression**. Otherwise, if a program instruction was lost, **the program would not be the same**. The **meaning** of a text document could also be **changed**.

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

.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 **lossy compression**. 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 **lossy 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 **lossy compression**.

Programs

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