Key information:

Card: 4.1

Key vocab	
Binary	Numbering system which uses base 2 (0s & 1s) – the only language that computers truly understand. 0 means off, 1 means on.
Denary	Numbering system which uses base 10 (0-9) – these are our normal numbers that we use every day. (Otherwise known as decimal)
Hexadecimal	Numbering system which uses base 16 (0-9 and A-F). These numbers are used to represent colours and code in assembly language, as they are easier for humans to understand than binary.

Binary shifts

Left hand shift will multiple a binary number by 2.
Each shift multiples the number by 2
1st shift is 2, 2nd shift is 4, 3rd shift is 8 etc.

Right hand shift will divide a binary number by 2.
Each shift divides the number by 2
1st shift is 2, 2nd shift is 4, 3rd shift is 8 etc.

Overflow

Understand that if the result of an addition or shift process results in a number that is too large to fit in the space available then an overflow has occurred.

Example=
When we add two numbers together that would produce a number over 255 (we would generate a 9th binary value)

Key information:

Card: 4.2

Denary to binary

Step 1	Draw the table below.
Step 2	Think?!- how many of the numbers at the top of the column can I use to make up the number I'm trying to convert?
Step 3	Place a 1 in the columns you can use and a 0 in the ones you can't use.
Step 4	Make sure all the columns with a 1 add up to the number you are converting.

Enter the number you want to convert here

#	128	64	32	16	8	4	2	1
27	0	0	0	1	1	0	1	1

- The numbers 128, 64 and 32 are too large
- I can use 16 to make up 27. I take away 16 from 27 and that leaves 11.
- I can use 8 to make up 11. take the numbers away from each other.
- That leave 3
- 4 is too large but 2 and 1 are ok. Add the columns with a 1 in and I get 27.

Key information:

Card: 4.3

Binary to denary

Step 1	Draw the table below.
Step 2	Think?!- how many of the numbers at the top of the column can I use to make up the number I'm trying to convert?
Step 3	Add up all of the columns with a 1 in them.
Step 4	Make sure all the columns with a 1 add up to the number you are converting.

Enter the number you have converted

#	128	64	32	16	8	4	2	1
130	1	0	0	0	0	0	1	0

- I have a 1 in two columns.
- I add the numbers with a 1 in their column

Key information:

Card: 4.4

Hexadecimal uses the numbers 0-9 and then for:

- 10 we use A
- 11 we use B
- Etc. until we reach the number 16 which is represented with an F

Binary/denary to hexadecimal	
Step 1	Draw the table below.
Step 2	If it's a denary number you are converting, put it into binary. <u>At this stage we want a binary number.</u>
Step 3	Add an additional row at the top and split the table into 2 sets of 4. place a 1,2,4,8 above the 16,32,64,128
Step 4	On each side of the new table add up the numbers. So in the example below we add the 1's on the left hand side. This adds up to 12. We know 12 is C so the first part of the answer is C.
Step 5	On the right hand side we do the same. We add up the 1's. we add the 4, 2 and 1. we get 7. We bring them together and get C7. This is our hexadecimal number to represent 99.

	8	4	2	1				
#	128	64	32	16	8	4	2	1
199	1	1	0	0	0	1	1	1

Key information: Hexadecimal uses the numbers 0-9 and then for:

- 10 we use A
- 11 we use B
- Etc. until we reach the number 16 which is represented with an F

Card: 4.5

Hexadecimal to binary/denary	
Step 1	Take your hexadecimal number and split it into its 2 parts. C7 we split into C and 7
Step 2	If the left hand side is a letter we convert it to its number. So C is 12 We do the same thing with the right hand side. If either is a number we leave it as it is
Step 3	We now have 2 numbers. We multiple the left hand number by 16. so in our example we multiply 12 by 16. We get 192.
Step 4	We add this new number to the number on the right. In our example we add 192 with 7. this is the final denary number
Step 5	If we want this hexadecimal number to be in binary we just plot it into the table we have used before.

Example 1

Hexadecimal number= A 5
Step 2= A is 10
Step 3= 10*16
Step 4= 160+5 (right hand number)
Final denary number= 165

Example 2

Hexadecimal number= B F
Step 2= B is 11, F is 16
Step 3= 11*16
Step 4= 176+16 (right hand number)
Final denary number= 192

Key information: Hexadecimal uses the numbers 0-9 and then for:

- 10 we use A
- 11 we use B
- Etc. until we reach the number 16 which is represented with an F

Card: 4.6

Understand that a binary number is far easier to use as the shorter hexadecimal notation.

Example 1

Binary number= 001110001111
Step 1= split each binary number into 4's
Step 2= convert each set of 4 hex values
Step 3= 0011 becomes 3, 1000 becomes 8 and 1111 becomes 15 or E
Final hexadecimal number= 3,8,E

Key information:

Card: 4.7

Binary addition	
Step 1	Draw the table. Setup your binary numbers as a normal addition calculation in two rows. Convert if needed. Convert your denary number to binary
Step 2	Add up the numbers.
Step 3	Place a number 1 in the columns that allow you to make up the added numbers

Example

Add= 45 and 00000001

#	128	64	32	16	8	4	2	1
45	0	0	1	0	1	1	0	1
1	0	0	0	0	0	0	0	1
46	0	0	1	0	1	1	1	0

Key information:

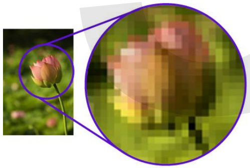
Card: 4.8

Vector images

1. These are made up of lines, shapes and curves NO PIXELS.
2. Their dimensions can be changed without affecting the quality of the image
3. They are smaller than bitmap images (memory size)

Bitmap images

1. These are photographs that are made up of pixels
2. Individual pixels can be edited
3. If their dimensions are changed this affects the quality of the pixels
4. Large file sizes



Raster images

1. Dot matrix of pixels to represent an image
2. Very large file size
3. If their dimensions are changed this affects the quality of the pixels

Sound

That sound is stored as a digital
representation. The digital representation is achieved by sampling (signal processing).
The sample quality can be affected by the sample rate and sample frequency. The higher the sample rate and frequency, the larger the resultant sample.

Key information:

Card: 4.9

Meta data

<u>Data about data</u>

Data stored about a particular file. Examples include:

Location
Author
Recording date
Artist
Song title etc.

Word Document Metadata Example

Key information:

Card: 4.10

Character sets
Characters, symbols, numbers are all represented by a unique binary code.

We use ASCII to be able to interchange data between different programs. This is called a characters sets. There are different versions

Unicode is another example of one. It can represent more characters but needs more memory to store its character set.

Data types
Integer (whole number) 7, 0, 15, -5
Boolean (true/false)
Real (number with, or without, fraction) 7.2, 8.9, -6.8, 12.0
Character (single character) a, @, #, 8, Q
String (one or more character) Hello, abc, b, Y

Data structures
Ways we can store and structure data (practical examples in class will be given)

Key information:

Card: 4.11

Validation
A way to make sure data is entered so it matches a rule (its valid)

Range check → Does it match a range
Examples= between 0-9, >3 and <9

Type check → Does it match a type expected
Examples= for age I would want an integer

Presence check → Has anything been entered?

Lookup table → Select from a list provided
Examples= M, F

Check digit → Does it have a digit in a particular place
Examples= JH379320A (the A is a check digit)