

# Basic background

*A class is a pattern for a set of similar objects that may contain data operators and actions.*

**A**n object is an instance of a class that lives in the computers primary storage, the computer memory. The computers secondary storage is normally the computer's hard disk. Think of this memory as a spreadsheet with cells as visualised in the figure below.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A																
B					1	1	1	1	1	0	1	0				
C																
D																
E																
F																
G																
H		1	1	1	1	1	0	1	0							
I																
J																
K																
L									1	1	1	1	1	0	1	0
M																
N																
O																
P																

In column E, row B, we refer to this as cell EB, we have put an one, then a new one in cell FB and so on, all in all 8 0's and 1's. We will call this a bit stream containing 8 bits. Another word for a bit stream containing 8 bits is a byte. This specific byte is an instance of an object. In this case it is a whole number or an integer, the number 250. This fictional computer has a memory of 16x16=256 cells. A computer is digital and it operates on 0's and 1's or bit streams. Another byte start in cell BH and yet another in cell IL. Studying the three bit streams, we observe that we have three identical versions of

the same bit stream or the same integer, in this case the whole number 250. We now know that a byte is 8 bits long. If we call the first byte starting in cell EB the source byte and the other two copies, we see that these two three bytes occupy 24 bits or cells of the total memory of 256 cells. But is it necessary to copy the first byte? Is there another way we can tell the computer the content of the byte starting in memory location, that is cell B H or IL. The answer is yes. The computer language we will use understands something called a reference and we use a so called reference operator, &. The language also understands a key word called int. So if we declare our first integer as int1, the second as int2 and the third as int3, we have the following relation:

$\text{int1}=\text{int2}=\text{int3}=250$  (decimal) = 11111010 (binary).

It is not difficult to find a **binary calculator** (Search Engine Key Word – SE KW) on the internet, where you can convert decimal numbers to binary and binary to decimal.

So long so good, back to our bit stream, the binary number int1 or 11111010, and its two copies int2 and int3. What happens if we change the last bit of int1, so  $\text{int1}=11111011$ , that is another decimal number, namely 251? The new number is now different from the other two. Int1 has changed value to 251, while  $\text{int2}=\text{int3}=250$ . By using the reference operator, we can secure that int2 and int3 changes when the source number changes value. So by defining  $\text{int2}=\text{int3}=\&\text{int1}$ , the content in the three bytes will always be the same. Another advantage is that a reference takes less space in the computers memory.