Here 12 is taken away from 16 but the process of subtraction can equally be seen as an addition: the addition of 16 and negative 12. In both cases the answer is the same and the only difference is the use of arithmetic signs and brackets. This slight modification can be used by the computer to both represent negative numbers and simplify the problem of subtraction.

For simplicity assume our computer is only large enough to handle 5 digits. Of course real computers can handle numbers with thousands of digits. Our 5-digit computer adopts a method of working: the leading digit on the left hand side is to be considered separately from the other 4. If the leading digit is 1 it is to represent negative 16 and if it is 0 then it is of course a zero. The remaining 4 digits are positive and follow the binary conventions we saw in the last instalment of the course

[]	[]	[]	[]	[]
−16 or 0	8s	4s	2s	1 or 0

So for example the binary number 01000 is decimal 8 and 10000 is decimal −16. But what about 10100? This includes −16 and +4 giving −12.

How many numbers can be represented with only 5 digits using this convention? The largest positive number is 01111 or decimal 15 and the greatest negative number is 10000 or −16. With a little experimentation you will see that every number between −16 and +15 can be represented.

Binary	**Decimal**
10000	−16
10001	−15
10010	−14
10011	−13
10100	−12
etc.	
11111	−1
00000	0
00001	1
00010	2
etc.	
01110	14
01111	15

If we increased the number of digits our computer was able to handle, we could of course expand the range of numbers.

Early on in the development of binary computer arithmetic, a very simple trick was discovered for finding the Two's Complement, or negative form, of a number. There are two steps to this trick.

First, invert each digit. So whenever you see a 1 put a 0 and wherever there is a 0 change it to a 1. Secondly, add 1 to the reversed number.

Follow the method as it is laid out in the example below. We are using +12, the binary equivalent being 01100. (The leading 0 on the left hand side is not strictly necessary as 01100 is the same as 1100. But since our computer has 5 digits we must fill up every one).

```
              01100  (= +12)
First Step:   10011
Second Step:  00001  (+1)
              10100  (= −12)
```

Now let's look at how our computer tackles a problem of subtraction; for example: 12 minus 4.

```
+12 is      01100
−4 is       11100 (using Two's Complement)
12+(−4)    101000
```

Notice that we now have 6 digits. Since our computer is only large enough to register 5 digits, the leading left hand digit is called an overflow digit and is ignored, leaving 01000 or 8 in decimal, the correct answer! A slightly more complex example is: 4 minus 12.

```
+4 is       00100
−12 is      10100
4+(−12)     11000
```

As a final example let's try dealing with two negative numbers together: −3 −4 = −3+ (−4) = −7

```
3 is         00011
so −3 is     11101 (using Two's Complement)
and −4 is    11100
            111001
```

Again we are left with a 6-digit number. Once the overflow is discarded, we have the binary number: 11001 or −7 in decimal.

These subtractions used only addition and the trick of the Two's Complement (which itself uses only reversal of digits and addition). The advantage to the computer is that binary digits can easily be reversed using a NOT Gate (see page 68).

A NOT Gate has one input and one output. It is a very 'perverse' Gate because whatever value you feed in, the output is the opposite. So if the input is 0 the output is 1 and if the input is 1 the output is 0. This characteristic of 'inverting' is exactly what is needed for the first (the inversion) step in the trick of the Two's Complement. In the next part of the course we will see how addition can easily be carried out by a computer using a combination of logical Gates.

Dots And Dashes
Morse code is one of the earliest illustrations of binary coding in electronics. In 1837 the first electric telegraph was laid in London with two miles of cable joining Euston to Camden Town railway stations. Later the same year in America, Samuel Morse demonstrated his celebrated code for transmitting messages. Each letter was a combination of two signals: dots and dashes