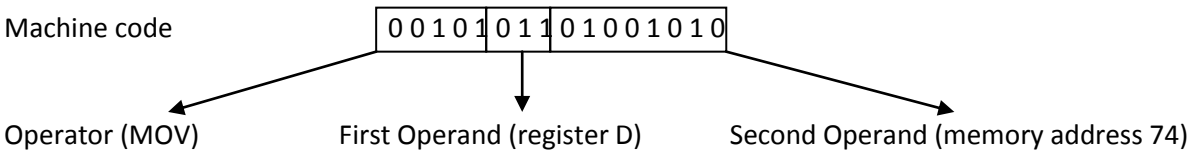
Coding Instructions - Reading

In Maths we show calculations like this: 5 + 4 = 9

The plus symbol is known as an **operator**. It performs its **operation** on two **operands**, in this case 5 and 4. In the same way, computers use operators and operands in **assembly and machine code**.

All instructions and data on a computer are stored in RAM. The computer itself doesn't know if a memory location is storing an instruction or data. All it knows is where the first instruction is that it needs to run. This is stored in a special register called the **program counter**. **Registers** are memory on the CPU that store 1 to 8 bytes of data. They operate at the same speed as the CPU.

The CPU first **fetches** an instruction in binary from the RAM. It then **decodes** this instruction to work out what operation it needs to do (the operator), and what the operands are (the data). Finally the CPU will **execute** the instruction. This is known as the **fetch-execute cycle**.



Assembly instruction: MOV D, [74]
Meaning: Move the contents of memory address 74 to register D

The program counter is now **incremented** (1 is added to it) so that it points to the location for the next instruction in memory.

An instruction set is a list of all the instructions that can be given to a particular CPU. This is specific to each type of CPU. This is why a program that has been compiled for one processor (e.g. a PC with a 64 bit processor) won't work on a computer with a different processor (e.g. a PC with a 32 bit processor).

An example instruction set is given below

Instruction	Assembly code example	Meaning
00000 XXX YYYYYYYY	SUB A, 73	SUBtract value in YYYYYYYY from register XXX
00001 XXX YYYYYYYY	ADD A, 125	ADD value in YYYYYYYY to register XXX
00010 XXX YYYYYYYY	MOV B, 6	MOVe value YYYYYYYY into register B
00011 XXX	DEC C	DECrement (subtract 1) from the value in register XXX
00100 XXX	INC D	INCrement (add 1) to the value in register XXX
00101 XXX YYYYYYYY	MOV A, [5]	MOVe value in memory address YYYYYYYY to register XXX

Question: What would the following machine code do?

Machine code	Assembly code	What it does (Answer)
00001 001 10110110	MOV B, 182	Puts the value **182** into register B
00001 000 01001001	MOV A, 73	Puts the value **73** into register A
00011 000	DEC A	Decrements register A which now stores **72**
00100 001	INC B	Increment register B which now stores **183**
00001 000 00100101	ADD A, 37	Add 37 to register A. Register A now stores 72+37=**109**