

# Computer Science 1MC3

## Lab 3 – Control Structures Part 1: If statements

---

In C there are two very important types of control structures, if statements and loop statements. We will cover if statements first.

The syntax of an if statement is as follows:

```
if (statement that evaluates to true or false) {  
    execute statement  
}  
else if (optional second statement) {  
    execute statement  
}  
else {  
    statement that will run if  
    all previous statements fail.  
    This is also optional.  
}
```

If statements allow you to have code run only under certain conditions. The addition of `else if` and `else` are used to clarify code. Without them you would have to use a lot of if statements to convey the same meaning.

You may be wondering what kind of statement results in true and false. Well these are mostly mathematical identities, such as less than or equality.

```
int x=4, y=8;  
  
if (x<y) {  
    printf("x<y is true");  
} else if (x>y) {  
    printf("x>y is true");  
}  
  
} else {  
    printf("x is equal to y");  
}  
}
```

The other type of control structure mentioned was looping. There are three different loops in C, for loops, do while loops, and while loops. For simplicity sake we will only cover the while loop in this weeks tutorial. The syntax for a while loop is as follows:

```
while (statement is true) {  
    execute statement  
}
```

You may also have a loop within a loop, this is known as nesting your loops.

```
int x=4, y=8;

while (x<y) {
    x++;
    while(y>x) {
        y=y-6;
    }
}
```

In c there are also a few symbols and short hands that you should be familiar with while writing and reading code.

x++; ° x=x+1;

x--; ° x=x-1;

x!=7 ° not equal to 7

**Questions:**

1) Trace the following code, provide your answers in the spaces:

```
int x=5, y=8, z=9;

while (x<z) {

    x++;
    trace(x, y, z);

    if (x==y) {

        x=z+y;
        trace(x, y, z);

    }

}

x=x/y;
trace(x, y, z);
```

	<b>x</b>	<b>y</b>	<b>z</b>
1			
2			
3			
4			
5			

2) Convert the code from question 1 into a flow chart. Where it says “trace” just put a box saying “trace”.

3) Convert the following flow chart into code:

*Andrew could you please do a flowchart for the following code, I was to lazy to do it myself and scan it. It would be great if it could fit into this space.*

```
int fox=2;
int rabbit=4;

while (rabbit>0) {
    rabbit=rabbit+2;
    rabbit=rabbit-fox;
}
```

*also here is the answers to the trace:*

<i>6</i>	<i>8</i>	<i>9</i>
<i>7</i>	<i>8</i>	<i>9</i>
<i>8</i>	<i>8</i>	<i>9</i>
<i>17</i>	<i>8</i>	<i>9</i>
<i>2</i>	<i>8</i>	<i>9</i>

*Thanks Andrew, I'm also going to send this off to Ian for proofing.*