Chapter 5 – Loops

For Loop

Let’s say you wanted to print out the numbers one to a hundred. What way would you do it?

Would you use one hundred lines like this:

```
cout<<"1"<<endl;
cout<<"2"<<endl; .....Etc.
```

This way would be very time consuming. A much quicker way would be to use a "for loop". For example, below we use a “for loop” to print out the numbers between one and a hundred.

```
#include<iostream.h>

int main()
{
    int i;
    for(i=0; i<100; i++)
    {
        cout<< i <<endl;
    }
    return 0;
}
```

Example 5a

When the “for loop” is first encountered, i is set to 0, and this is printed out with the cout statement. Next time around i is incremented by one (i++). Now i equals 1, and this is printed out. Then i is incremented again and the process continues for as long as i is less than 100 (i.e. the program will print out 0 to 99).

Note that i++ is a shorter way of saying i = i+1. In maths this statement does not make sense. If i was 10, then this statement would be saying 10=10+1 or 10=11. We obviously know this to be untrue, but in programming it means take the “old i” and add 1 to it to get the “new i”.
Another example:

```cpp
#include<iostream.h>

int main()
{
    int i;

    for(i=0; i<=100; i++)
    {
        cout<< "hello!" <<endl
    }

    return 0;
}
```

**Example 5b**

This piece of code will print out “hello” 101 times on your screen. The 101 is because i starts at 0 and continues until it is less than or equal to 100 (i<=100).

Try out the following for loops in your code.

```cpp
for(i=100; i>=0; i--) //Note: i-- means i = i - 1
{
    cout<< i <<endl;
}
```

```cpp
for(i=0; i<100; i+=2) //Note: i +=2 means i = i + 2
{
    cout<< i <<endl;
}
```
Do While Loop

If you do not know how many iterations it will take to do something then you use a “do while” loop.

For Example:

```
#include<iostream.h>
int main()
{
int x;

do
{
    cout<< "enter a number" <<endl;
    cin>>x;
}
while(x!=0); //While x is NOT equal to 0
return 0;
}
```

Example 5c

This program will keep asking the user to enter a number unless 0 is entered. The program exits the “do while” loop when 0 is entered.

Note: x!=0 means x NOT equal to zero