Functions- Part2
Lecture 2

University of Anbar

College of Computer Science and Information Technology

Department of Computer Science

Object Oriented Programming

Second Class

Dr. Ruqayah R. Al-Dahhan

Outlines:

-The conditional operator

-Functions:

- default parameters

The conditional operator

A 'ternary' operator - it takes three arguments:

```
    expr1 ? expr2 : expr3
```

If `expr1` is `true` then `expr2` is evaluated otherwise `expr3` is evaluated.

Remember logical expressions evaluate to a true or false

The conditional operator

It allows a shorthand form of an `if` statement:

```
if (y < z) {
  x = y;
} else {
  x = z;
}
```

can also be written:

```
x = (y < z) ? y : z;
```

Sometimes this makes for more compact or efficient code.

Example

```
#include <iostream>
using namespace std;


main(){

int x,y;
cout<<"Enter two integer numbers";
        cin >> x;
        cin>>y;

        int z;
/*if (x>=y)
        z=x+7;

else
        z=x+y;
*/
z= (x>=y) ? x+7:  x+y;

cout<<" the value of x= "<<x<<endl;
cout<<" the value of y= "<<y<<endl;
cout<<" the value of z= "<<z<<endl;
}
```

Functions

As a matter of style, every function ought to have its prototype declared before `main()` and its definition after `main()`.

At any rate functions **must** have at least their prototypes set up before they are used. So the compiler recognises their signatures. (signature = type, name and argument number and types)

In C++ the use of `void` is optional in a function that has no arguments:

```
int fct() { return global*2;  }
```

Instead of :

```
int fct( void ){return global*2;  }
```

Default parameters

```
int power(int n, int k = 2) {
                //k has a default value of 2
   if (k == 2) { return n*n; }
   return power(n,k-1)*n;
}
```

we can call this function in two different ways:

```
j = power(5,3); // 5 cubed
j = power(5);   // 5 squared
```

Default parameters

Remarks:

Only trailing arguments may be defaulted.

More than one argument can be defaulted:

```
    void fct(int j=4, int k=5, int m=7);
```

So we can call

`fct();` or `fct(1);` or `fct(1,2);` or `fct(1,2,3 );`

Question

What is the output?

```
#include<iostream>
using namespace std;
 main(){
     int x=65, y=23, H;
  H=(x>y)?x+4:y-3;
  cout<<H;
}
```