

# C++ part 3

Ned Lowe

ned.lowe@ic.ac.uk

*“Recursion (ri-'k&r-zh&n) [noun] - See:  
Recursion”*

# Functions

- Functions allow us to take a collection of code and group it together away from the main program
- Function is passed parameters, then returns a result
- Makes code more readable, more elegant and allows re-usability

# Using functions

- Give each function a name
  - When we want to use it we ‘call’ function by it’s name
- Any variables which we want to be available to the function need to be ‘passed’
- The result (returned value) needs to have a type
- If no returned value is needed then declare the type as ‘void’

# Creating a function

- Two parts
  - function *prototype*
  - actual function
- function prototype tells the compiler about the function
  - parameters to pass
  - return value

# Prototype

- *type funcName(param1Type param1Name, param2Type param2Name);*
- `int Area(int Width, int Length);`
- `bool Validate(char UserInput);`
- `void PrintMessage(int MessageNumber);`
- Place prototypes for all functions in a list at beginning of your program, after `#include` statement

# Actual function

- First line is almost the same as prototype
- Rest is the same as 'main' function
- Value to return is placed after the 'return' keyword

```
type funcName(param1Type param1Name, param2Type param2Name) {  
    statements  
}
```

```
int Area(int Width, int Length) {  
return Width*Length;  
}
```

# Use in a program

- Write the name of the function
- In the brackets put the values you want to pass to the function
  - These values 'become' the parameters in the function

- ```
int UserWidth=10;  
int UserLength=20;  
int UserArea;
```

```
UserArea = Area(UserWidth,UserLength);  
cout << UserArea << endl;
```

```
#include <iostream.h>
int Area(int Width, int Length);

int main()
{
    int UserLength;
    int UserWidth;
    int UserArea;
    cout << "Please enter the length\n";
    cin >> UserLength;
    cout << "Please enter the width\n";
    cin >> UserWidth;
    UserArea=Area(UserLength,UserWidth);
    cout << UserArea << endl;
return 0;
}

int Area(int Width, int Length) {
return Width*Length;
}
```

```
#include <iostream.h>
void PrintMessage(int MessageNumber);

int main() {
    int UserNum = -1;
    while (UserNum < 0) {
        cout << "Please enter a positive number\n";
        cin >> UserNum;
        if (UserNum < 0)
            PrintMessage(1);
        else
            PrintMessage(2);    }
    return 0;
}

void PrintMessage(int MessageNumber) {
    switch (MessageNumber) {
        case 1: cout << "That number was negative. Stupid!\n"; break;
        case 2: cout << "That number was positive. Thankyou!\n"; break;
    }
}
```

# Global and Local Variables

- Any variables created in a function are ‘local’
  - When the function ends, the variable disappears
- When you pass a parameter it creates a new variable with the same value as the variable you pass
  - The original variable remains unchanged
  - We will see a way to change the original later
- ‘Global’ variables are available to all functions
  - They are bad! We won’t use them.

# Returning values

- Once a function has returned a value, it ends

```
int Triple(int PosNumber) {  
    if (PosNumber <= 10,000)  
        return PosNumber*3;  
    else  
        return -1;  
    cout << "You will never see this\n";  
}
```

```
#include <iostream.h>
float Inchtocm(float OriginalNumber);
float CmtotInch(float OriginalNumber);

int main() {
    float UserNum;
    float Answer;
    int UserChoice = 0;
    cout << "Please enter the number you wish to convert\n";
    cin >> UserNum;
    while ((UserChoice!=1) || (UserChoice!=2)) {
        cout << "Please enter (1) if this is in cm, (2) if this in inches";
        cin >> UserChoice;
    }
    if (UserChoice = 1)
        Answer = CmtotInch(UserNum);
    else
        Answer = Inchtocm(UserNum);
```

```
cout << "The converted answer is " << Answer << endl;
}
return 0;
}
```

```
float Inchtocm(float OriginalNumber) {
return OriginalNumber*2.54;
}
```

```
float Cmtoinch(float OriginalNumber) {
return OriginalNumber/2.54;
}
```

# Example programs to create

- Take a number from user and draw a square of side length of that value
- Write a program which generates the Fibonacci Sequence
- Write a program to work out square roots of a quadratic equation given by user entered co-efficients