Programming in C++

"If programming in Pascal is like being put in a straightjacket, then programming in C is like playing with knives, and programming in C++ is like juggling with chainsaws"

Anonymous.

A First C++ Program

```
#include <iostream.h>
int main(){
  cout << "Hello World" << "\n";
}
```

Output:

```
Hello World
```

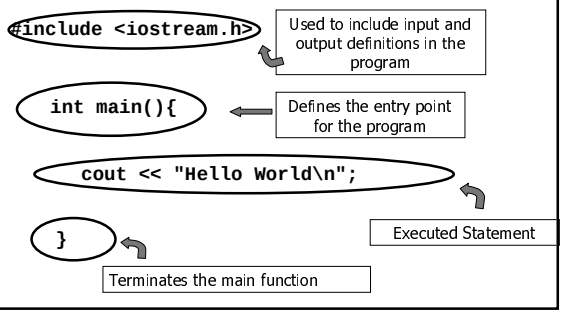

Console Applications

- This program is an example of a ***Console application***
- Console I/O functions perform
  - input from the system's standard input device ***stdin*** (usually a keyboard)
  - and output to the system's standard output device: ***stdout*** (usually a screen)
- Standard input and output may be redirected to other devices.
- MS-DOS applications were typically console applications

Console Subsystem

- Pure Windows applications do not use the Console I/O functions **`cin`** and **`cout.`**
- Such applications are ***event-driven***.
- However, Windows provides a console subsystem, which allows the emulation of an MS-DOS window for character-based applications.

A First C++ Program



- **`main()`** (or for Windows: **`winmain)`** is a function which is the entry point for every C++ program.
- **`{`** and **`}`** are used to bracket the body of the function **`main().`**
- The expression sends the string **`"Hello World"`** followed by a newline to the current output stream **`cout`**.
- **`cout`** is an *object* which is normally "attached" to the terminal. (It could also be attached, say, to a printer).
- You may also use **`cout`** in this way:

```
cout << "1+2+3 is: " << 1+2+3 << "\n";
```

Return Values

- Functions in C++ normally return a value when they finish
- If you do not wish for any return value, you must specify the void keyword, i.e. **`void bob()`**
- Alternatively, you could specify:

```
int bob(){
        … do something …
   return 0;     //Successful completion
}
```

Libraries

- Many programming tasks are routinely performed, e.g. math functions, I/O, graphics, memory management etc.
- C++ has a relatively small set of keywords, and all other functions are provided in ***libraries.***
- Library functions are not part of C++, but all compilers provide certain library functions.
- Programmers can also create their own libraries, or use those provided by third parties (e.g. ***OpenGL***).

Header Files

- Libraries are usually not provided in source form, but in compiled form
- When you build (or link) your program, these functions are combined with your compiled code, to produce the final executable file.
- Most library functions will need some definitions contained in particular files which are supplied with the system, called ***Header files***: **`*.h`**
- These files must then be ***included*** in your program, using the pre-processor directive **`#include`**

The pre-processor

- C++ source code is **compiled** (by a compiler), producing object code (*.obj)
- Then all the object code for each separate part of the program, include library functions, is linked together to produce the final executable code (*.exe)
- C++ has a ***pre-processor*** which processes the code before compilation.
- The line **`#include <iostream.h>`** is a directive to the pre-processor to replace this line by the contents of the file iostream.h

Comments

- Full line comments are as follows:
  `/* This is a full line comment */`
- Comments on a line of code are as follows:
  `cout <<"Hi";` ***`//output Hi to screen`***
- Several lines of comments:

```
/*   This program is the best
*    program I've ever written
*/
```

(**Note:** `/**/` may not be *nested*)

```
#include <iostream.h>
int main(){
   int countdown=10;
   int number, answer=0;
   while ( countdown >0) {
     cout << "\n Input a number: ";
     cin >> number;
     if (number < 0)
          cout << "\n No negative numbers!";
     else {
          answer+=number;
          countdown--;
     }    //end if
   } //end while
   cout << "\n The answer is: " << answer;
} //end main
```