

```
1
2 #include <stdio.h> ..... compiler directive
3
4 void f(void); ..... function prototype
5
6
7 int main(void) ..... entry point for program
8 { ..... braces define body of program
9   int i; ..... object declaration: variable
10
11   for(i=0; i<10; i++) f(); ..... for(initializer; test; increment)
12   ..... loop calls function 'f()' with no calling arguments'
13
14   return 0; ..... returns to operating system
15 } ..... closing brace ends body of program
16
17
18 void f(void) ..... function definition
19 {
20   int j = 10; ..... object declaration: variable with initialization
21
22   printf("%d ", j); ..... print text, % is a format specifier for a decimal integer, value of 'j'
23
24   j++; /* this line has no lasting effect */ ..... increment variable 'j' and comment
25 }
26
```

```

1
2  #include <stdio.h> ..... compiler directive
3
4  void f(void); ..... function prototype
5
6
7  int main(void) ..... entry point for program
8  { ..... braces define body of program
9    int i; ..... object declaration: variable
10
11    for(i=0; i<10; i++) f(); ..... for(initializer; test; increment)
12                                ..... loop calls function 'f()' with no calling arguments'
13
14    return 0; ..... returns to operating system
15  } ..... closing brace ends body of program
16
17
18  void f(void) ..... function definition
19  {
20    int j = 10; ..... object declaration: variable with initialization
21
22    printf("%d ", j); ..... print text, % is a format specifier for a decimal integer, value of 'j'
23
24    j++; /* this line has no lasting effect */ ..... increment variable 'j' and comment
25  }
26

```

The form of a C Program

All C programs will consist of at least one function, but it is usual (when your experience grows) to write a C program that comprises several functions. The only function that has to be present is the function called **main**. For more advanced programs the **main** function will act as a controlling function calling other functions in their turn to do the dirty work! The **main** function is the first function that is called when your program executes.

C makes use of only 32 [keywords](#) which combine with the formal syntax to form the C programming language. Note that all keywords are written in lower case - C, like UNIX, uses upper and lowercase text to mean different things. If you are not sure what to use then always use lowercase text in writing your C programs. A keyword may not be used for any other purposes. For example, you cannot have a variable called **auto**.

The layout of C Programs

The general form of a C program is as follows (don't worry about what everything means at the moment - things will be explained later):

```
compiler directives
function prototypes
global declarations

main()
{
    local variables to function main ;
    statements associated with function main ;
}

f1()
{
    local variables to function 1 ;
    statements associated with function 1 ;
}

f2()
{
    local variables to function f2 ;
    statements associated with function 2 ;
}
.
.
.
etc
```

Note the use of the bracket set () and {}. () are used in conjunction with function names whereas {} are used as to delimit the C statements that are associated with that function. Also note the semicolon - yes it is there, but you might have missed it! a semicolon (;) is used to terminate C statements. C is a free format language and long statements can be continued, without truncation, onto the next line. The semicolon informs the C compiler that the end of the statement has been reached. Free format also means that you can add as many spaces as you like to improve the look of your programs.

For more information:

<http://www.le.ac.uk/cc/tutorials/c/cccstrt.html>