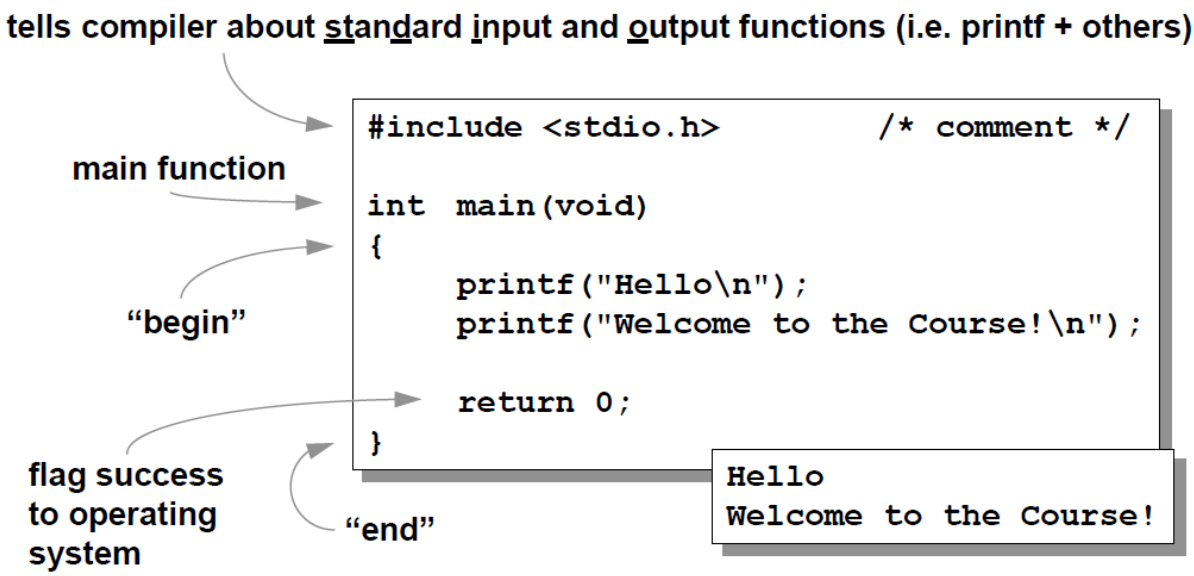

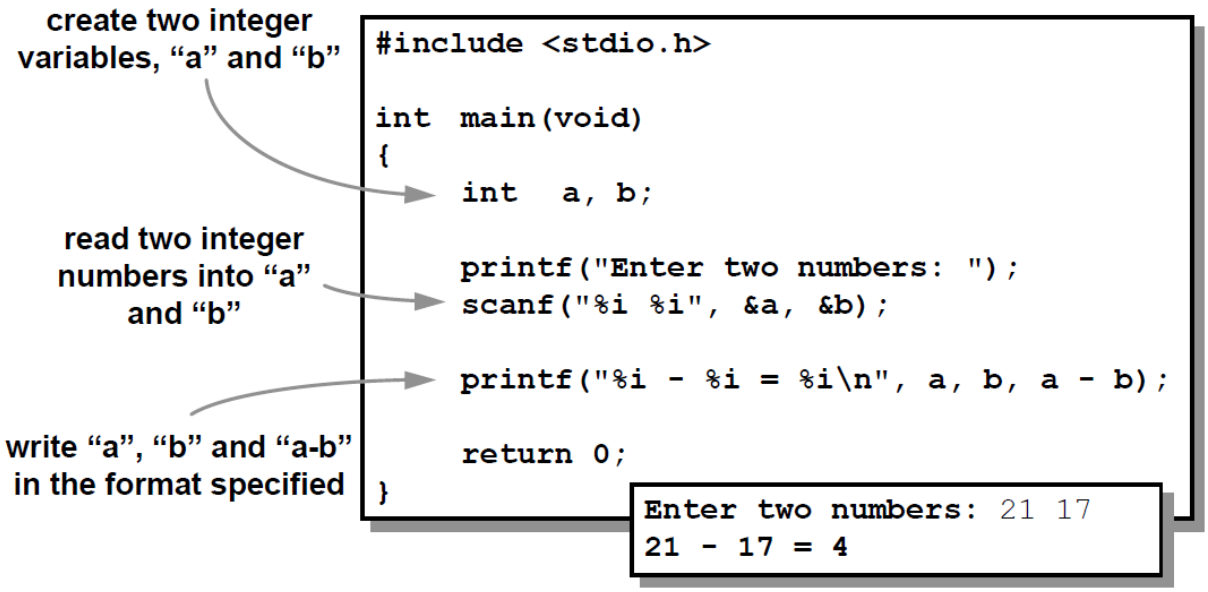

#include	The #include directive instructs the C Preprocessor (a non interactive editor which will be discussed later) to find the text file "stdio.h". The name itself means "standard input and output" and the ".h" means it is a header file rather than a C source file (which have the ".c" suffix). It is a text file and may be viewed with any text editor.
comments	Comments are placed within /* and */ character sequences.
main	The main function is most important. This defines the point at which your program starts to execute. If you do not write a main function your program will not run (it will have no starting point). In fact, it won't even compile.
braces	C uses the brace character "{" to mean "begin" and "}" to mean "end". They are much easier to type and, after a while, a lot easier to read.
printf	The printf function is the standard way of producing output. The function is defined within the Standard Library, thus it will always be there and always work in the same way.

\n	The sequence of two characters "\" followed by "n" is how C handles new lines. When printed it moves the cursor to the start of the next line.
return	return causes the value, here 0, to be passed back to the operating system. How the operating system handles this information is up to it. MS-DOS, for instance, stores it in the ERRORLEVEL variable. The UNIX Bourne and Korn shells store it in a temporary variable, $?, which may be used within shell scripts. "Tradition" says that 0 means success. A value of 1, 2, 3 etc. indicates failure. All operating systems support values up to 255. Some support values up to 65535, although if portability is important to you, only values of 0 through 255 should be used.
scanf	The scanf function is the "opposite" of printf. Whereas printf produces output on the screen, scanf reads from the keyboard. The sequence "%i" instructs scanf to read an integer from the keyboard. Because "%i %i" is used two integers will be read. The first value typed placed into the variable "a", the second into the variable "b". The space between the two "%i"s in "%i %i" is important: it instructs scanf that the two numbers typed at the keyboard may be separated by spaces. If "%i,%i" had been used instead the user would have been forced to type a comma between the two numbers.