

Lab 1:

Introduction to C Programming

(Creating a program using the Microsoft developer Studio, Compiling and Linking)

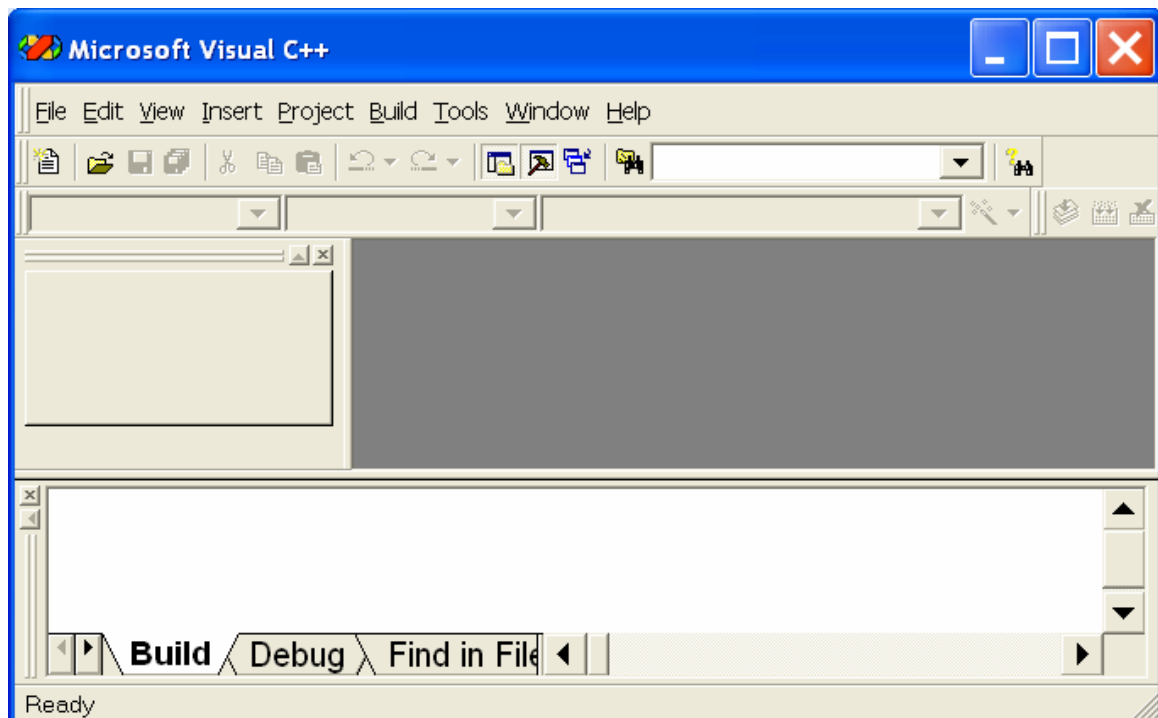
Learning Objectives

- 0. To become familiar with Microsoft Visual C++ 6.0 environment
- 0. To be able to write simple computer programs in C
- 0. To be able to use simple input and output statements

Lab Work

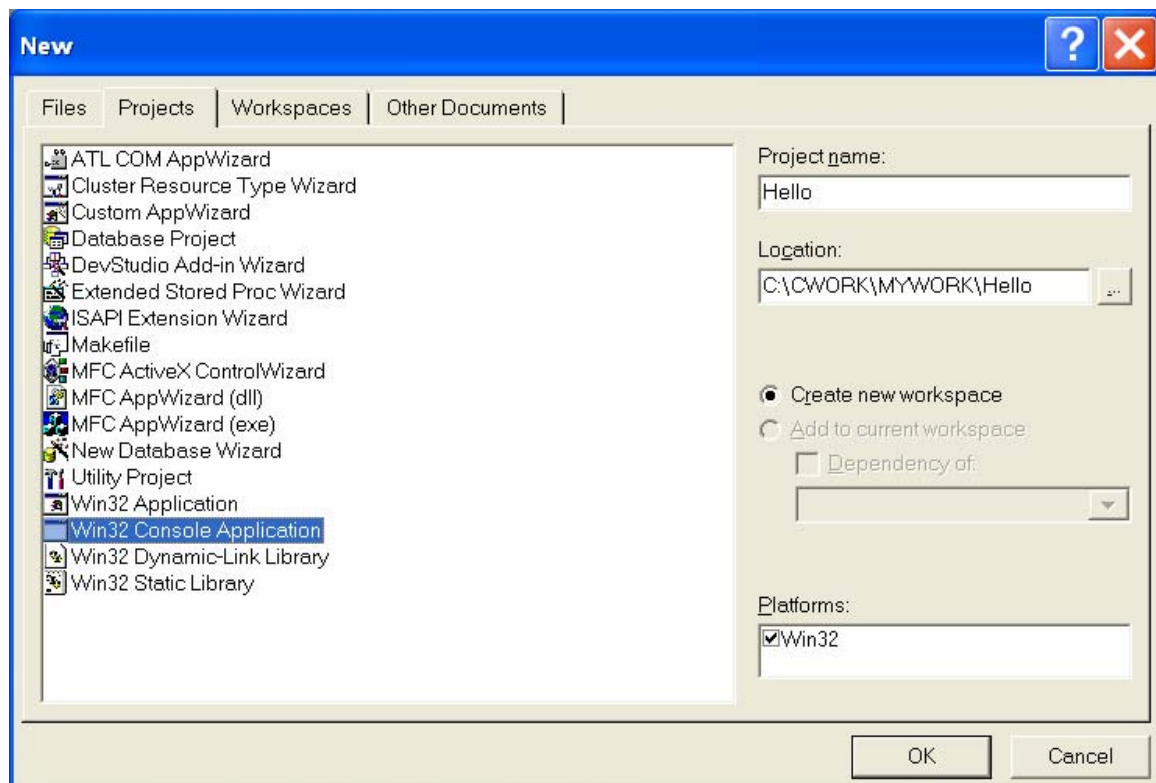
Part A: Creating the Project

1. Run the Microsoft Visual C++ 6.0 environment (Developer studio). You will see the following window.

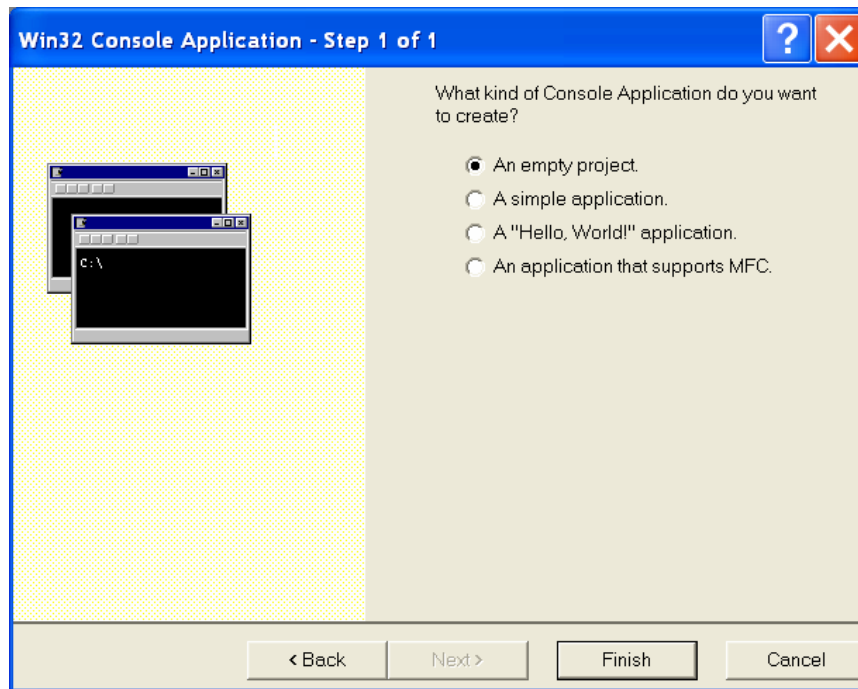


Once the Developer studio begins running, the first step is to create a project for the program you are about to write. A visual C++ project stores all the information required to build a particular program. This information includes the names and relationships of the program source files; a list of the required library files; and a list of all options for the compiler, linker and other tools used in building the program.

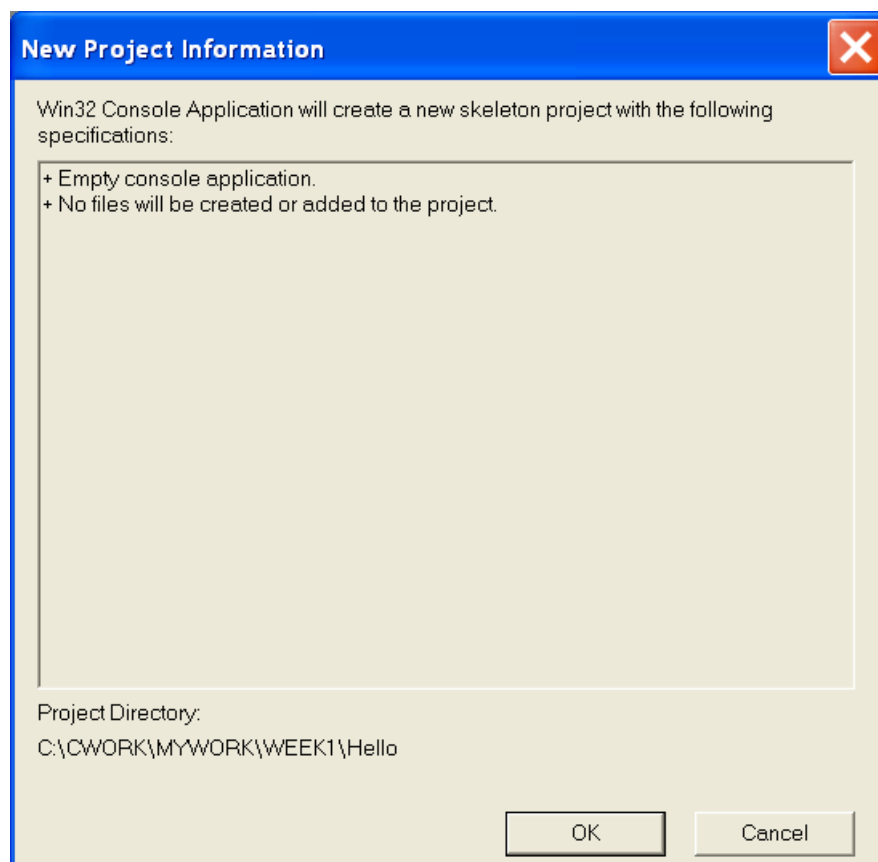
2. Choose the File → New ...menu command in the Developer studio or simply press Ctrl+N. The **New** dialog box will appear:



3. Open the Projects tab of the **New** dialog box (if it is not already open), so that you can create a new project.
4. In the list of project types, select the 'Win32 Console Application' item. A console application is a 32-bit character mode program.
5. Enter 'Hello' into the Project name: text box. This will cause the Developer studio to assign the name 'Hello' to the new project (as well as to the project workspace that contains this project).
6. In the Location: text box, specify the path of the folder that is to contain the project file(s), which is known as the *project folder*.
7. Click the OK button in the **New** dialog box. The Developer studio will now display the Win32 Console Application wizard, which consists of only a single dialog box.
8. In the Win32 Console Application dialog box select the 'An empty project' option as in the figure below. The option creates a project without adding any source code.

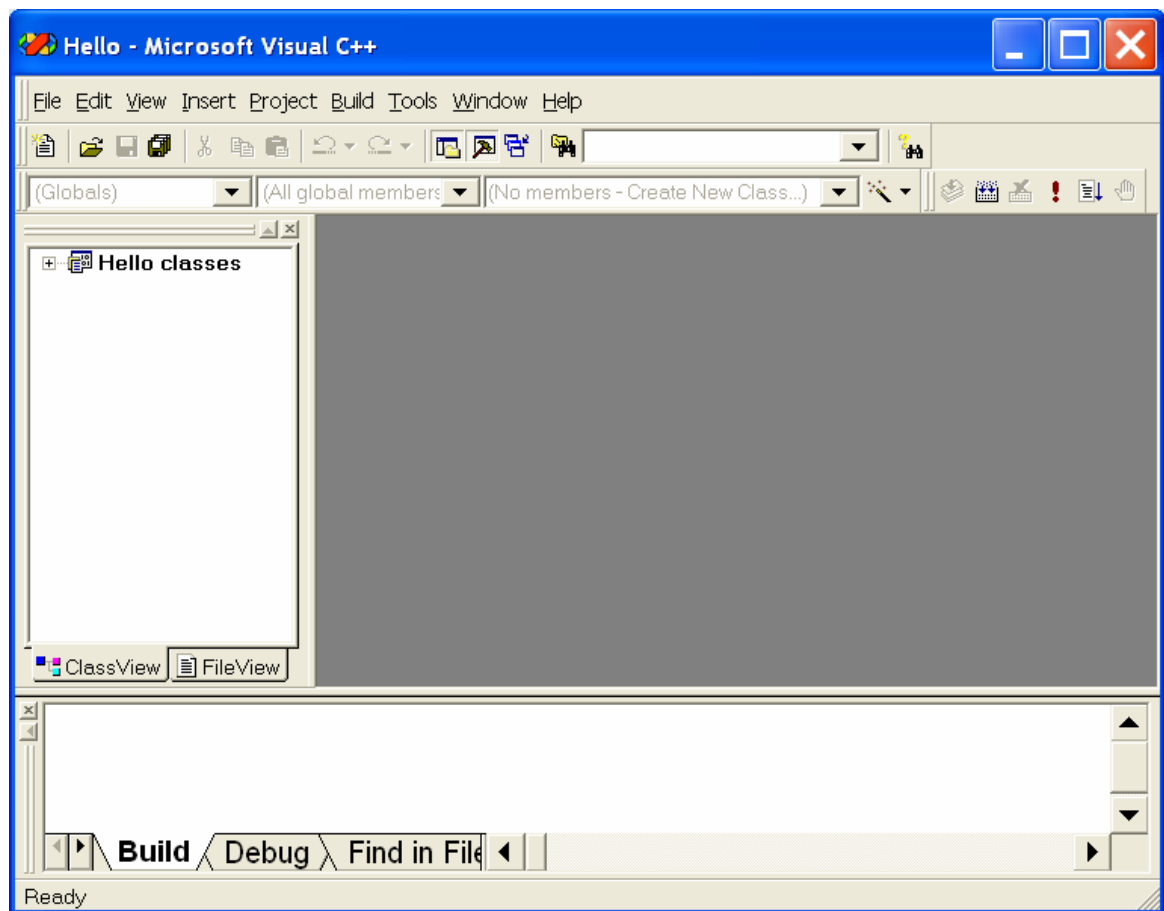


9. Now, click the Finish button at the bottom of the Win32 Console Application dialog box and the New Project Information dialog box will appear, which displays basic information on the new project that will be created. See Figure below.



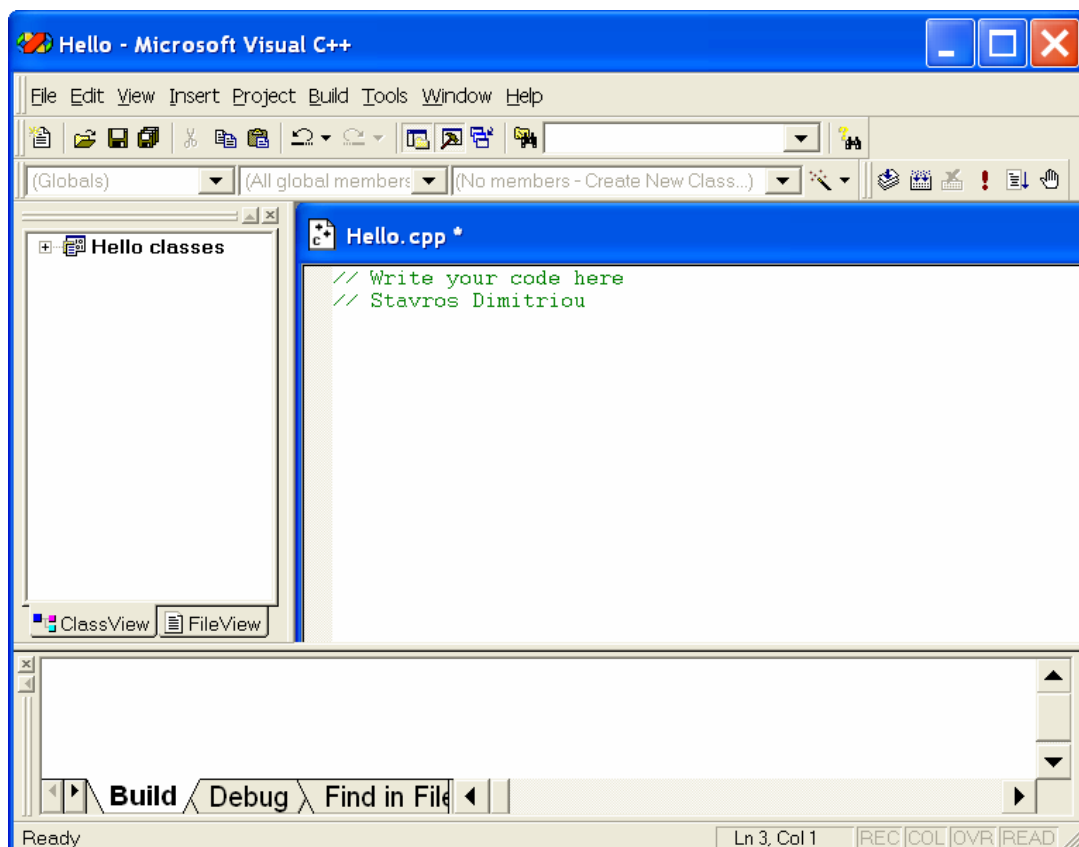
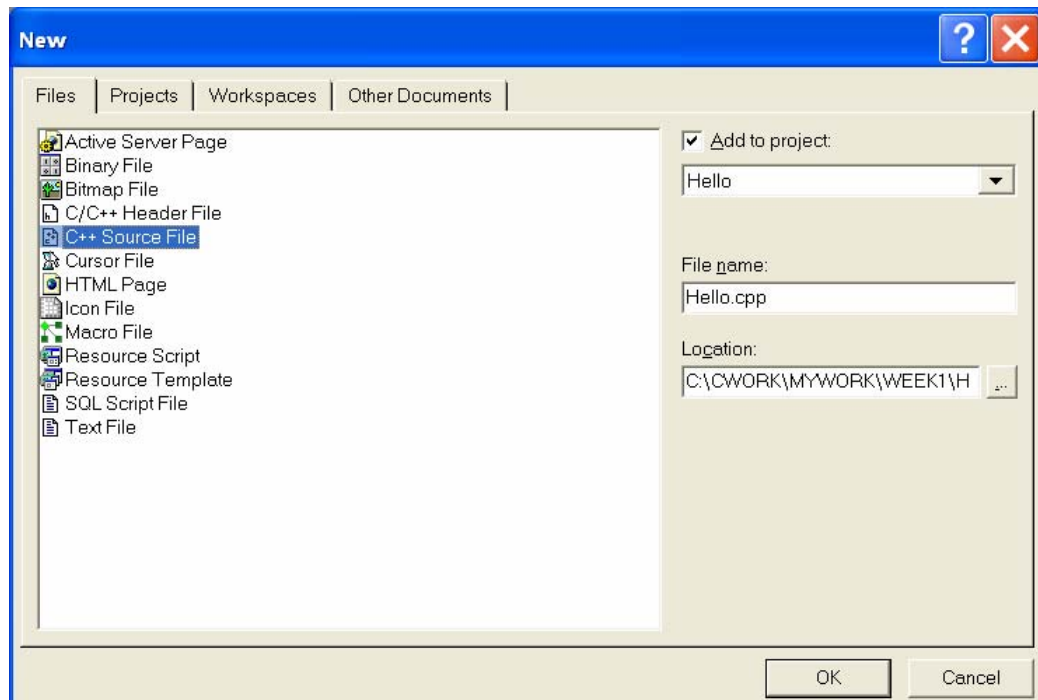
10. Click the OK button of the New Project Information dialog box. The Developer studio will now create and open a new project workspace, named Hello, which contains a single project, also named Hello. Information on the project will be

displayed within the Workspace window, as shown below:



Part B: Creating and Editing the Program Source File

1. The next step is to create the program source file, **Hello.c** for a C program or **Hello.cpp** for a C++ program. To do this, choose the File → New ...menu command in the Developer studio. Then, open the Files tab and select 'C++ Source file' item from the list of the file types. Then, enter the source file name, **Hello.cpp**, into the file Name: text box, and make sure that the Add To Project box is checked. The list under the Add To Project check box will already contain the name of the project, Hello, and the Location: text box will contain the full path of the project folder you specified when you created the project; leave both of these items as they are. The completed dialog box is shown in the figure below:
2. Click the OK button when you are done.



3. You are now ready to work on your C/C++ programs.

Part C: My First C Program

The C language facilitates a structured and disciplined approach to computer program design. C uses some notations that may appear strange to people who have not programmed computers before. We begin by considering a simple C program in order also make you understand how to edit, **compile** and **execute** the program using the Developer studio as explained in previous parts.

Exercise 1

1. Analyze the structure of following Java program “**Hello.c**”.
2. Understand the program and predict the outcome in your logbook.

```
/* Stavros Dimitriou: Hello.c
   A first program in C */
#include <stdio.h>

/* function main begins program execution */
int main( void )
{
    printf( "Hello World and Welcome to C!\n" );

    return 0; /* indicate that program ended successfully */
} /* end function main */
```

3. Create, compile and Run the program. Explain and analyse it in your logbook.
4. You are expected to get the following information after you have compiled the Hello.c program:

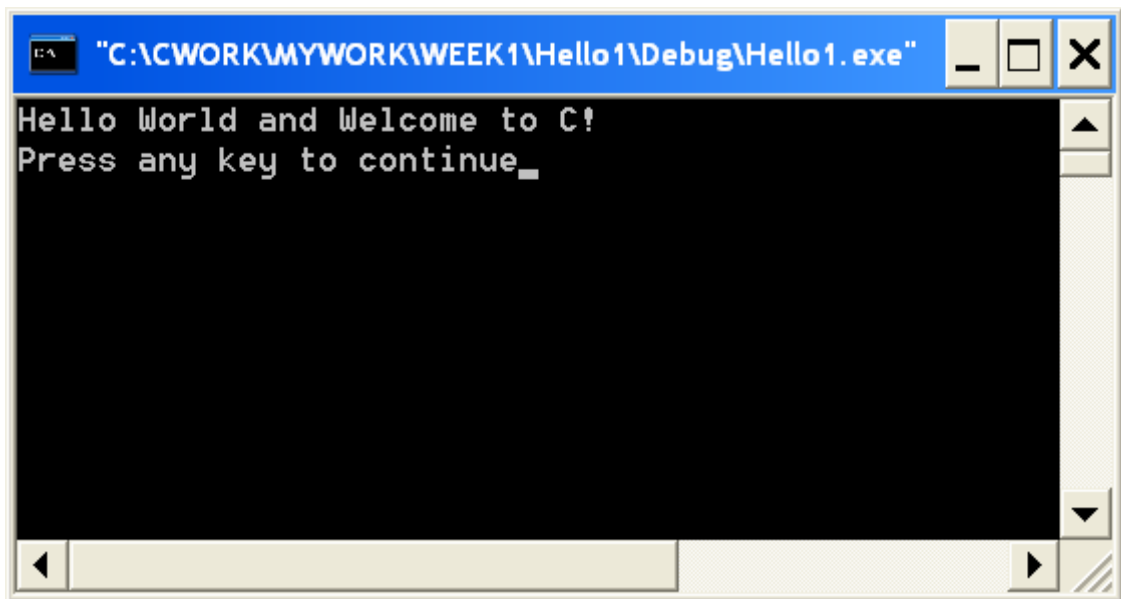
-----Configuration: Hello1 - Win32 Debug-----

Compiling...

Hello.c

Hello.obj - 0 error(s), 0 warning(s)

5. You are expected to get the following Output - as shown in Figure below - after you have successfully executed the Hello.c program:



6. Similarly work now on the following exercises.

Exercise 2

1. Analyze the structure of following Java program “**Welcome1.c**”.
2. Understand the program and predict the outcome in your logbook.

```

/* Stavros Dimitriou: Welcomel.c
   Printing on one line with two printf statements */
#include <stdio.h>

/* function main begins program execution */
int main()
{
    printf( "Welcome " );
    printf( "to C!\n" );

    return 0; /* indicate that program ended successfully */
} /* end function main */

```

3. Create, compile and Run the program. Explain and analyse it in your logbook.

Exercise 3

2. Analyze the structure of following Java program “**Welcome2.c**”.
5. Understand the program and predict the outcome in your logbook.

```

/* Stavros Dimitriou Welcome2.c
   Printing multiple lines with a single printf */
#include <stdio.h>

/* function main begins program execution */
int main()
{
    printf( "Welcome\nto\nC!\n" );

    return 0; /* indicate that program ended successfully */
} /* end function main */

```

3. Create, compile and Run the program. Explain and analyse it in your logbook.