1 Getting Started

The objective of this short description is to get you started with the Visual C++ environment. Before we proceed, it would be worthwhile to understand a few basic concepts regarding developing programs in VC++.

1.1 Basic Concepts

A workspace in VC++ refers to the ‘work area’ in which different projects and associated files are kept. Each workspace contains one or more projects. However, it is advised that one workspace should contain one project only unless there are more than one target files (executable files) that have to be generated. A project refers to a group of files related to one overall task. Again, a project can and normally does, contain more than one file. There are three broad categories of files in VC++ that form a project – source files (.cpp and .c), header files (.h) and resource files (.rc). In this course, we will not be dealing with resource files.

Now let us get started with using the VC++ environment. To launch the environment, go to Start → Programs → Microsoft Visual Studio 6.0 → Microsoft Visual C++ 6.0.
1.2 Setting up the environment

To run our program we need to have an *active* workspace and project. The term *active* merely indicates that we should have an open workspace in which a project has been initialized so that we can add/create our source file and run it.

We can create a workspace (and associated project) manually or use the default workspace (and associated project).

1.2.1 Manually Creating a Workspace

To manually create your workspace, click the ‘File’ drop-down menu and select ‘New’.

The following dialogue box will appear on your screen.

![Workspace creation dialogue box](image.png)

This dialogue box will allow you to create *Files*, *Projects*, *Workspaces* and *Other Documents*. The four tabs appearing at the top left of this dialogue box will allow you to create these different types of documents.

Notice that by default, the *Projects* tab is active and there is a long list that shows you the different types of Projects that can be created.
Introduction

Select the Workspaces tab and you will notice that a Blank Workspace option appears below the tabs. This indicates that you can only create a blank workspace through this dialogue box. Move your cursor to the Workspace Name text box that appears on the top-right corner of this dialogue box and type an appropriate name for your workspace. Just below this point is the Location text box where you should choose an appropriate path/directory for your workspace. You can type out a path of your choice or ‘select’ a path by clicking on this button that appears at the end of the Location dialogue box.

Once you have typed a name and chosen a path for your workspace, press the OK button. A vertical window will appear on the left-hand side of your screen:

The screen dump on the left-hand side indicates that a Blank Workspace entitled ‘Adeel’ has been created and right now it contains ‘0 project(s)’.

Now let us create a project inside this workspace. Right-click on the name of the workspace that appears in this window and four options will appear. Select the first one which says Add New Project to Workspace . . .

The dialogue box that will now appear is the same one which you just used to create the Blank Workspace.

As before, the Projects tab has been selected by default. There is a long list of different types of projects that can be created.

We are concerned with creating a Win32 Console Application. Therefore, select this third-last option and once it has been selected, type an appropriate name for your project in the Project Name text box that appears on the right-hand side. Make sure that you create this project ‘inside’ your new workspace. For instance, if your workspace entitled ‘myworkspace’ has been created at D:\MyFiles\, then you should create this project at this location: D:\MyFiles\myworkspace\.

When you press OK, another dialogue box will appear which will ask you to specify the type of project you would like to create.
Selecting the first option will allow you to create an empty project in which you will have to add your own files.

The second option will create some standard cpp and h files that will just form the skeleton of your intended program. For instance, if your project name is ‘firstproject’, then with the selection of the second option, VC++ will automatically create three files for you – StdAfx.cpp firstproject.cpp and StdAfx.h

All these files will have some basic statements defining the skeleton of your program. You may choose to create a project in this way, delete the StdAfx.cpp and StdAfx.h files and create your own files. As the name of this second option suggests – it is just a *simple application*.

The third option in the dialogue box (shown above) will allow you to create another ‘simple application’ with the added feature that if you run it, it will simply print ‘Hello World!’ on your screen. This option will also create those standard cpp and h files (StdAfx) mentioned above.

You need not concern yourself with the fourth option for this course as it deals with Microsoft Foundation Classes – a concept that will be dealt with in a higher level course.

Now, if you choose the first option – *An empty project* – from this dialogue box then another dialogue box will appear giving the features of the project you have chosen to create:
Note: The inset of this dialogue box shows that an empty project is being created. Had a simple application been created, then the names of the various files being automatically created would have appeared here as well. In addition, notice the path appearing at the bottom of the dialogue box. Here, ‘Adeel’ is the empty workspace that was created and ‘firstproject’ is the name of the empty project that has been created.

Once you press OK, no new window will appear but some additional information will appear in the window that gave you information about the blank workspace which you had created. Here is what the window should now look like:
Introduction

The workspace ‘Adeel’ which was empty earlier, now indicates that it contains one project. The name of the project appears in **bold** below the name of the workspace (firstproject in this case).

Double-clicking the name of the newly created project will show three separate directories, indicating the different types of files you can add to your project (Source, Header and Resource Files).

Right now, these directories will be empty. Proceed by right-clicking the source files directory. Then select the option *Add Files to Folder...*

A new smaller window will appear at the centre of the screen. Type the name of the new .cpp or .c file you wish to create in the *File Name* text box (Ex: file.cpp)

When you press *OK*, another prompt will appear telling you that the file name which you have specified does not exist. Press *Yes* to create the new file. Once you have done that, double-clicking on the *Source Files* directory will show you that it now contains the new file which you have created.

Double-click on the icon for your new file and a blank window will appear on your right. You are now ready to start coding!¹

1.2.2 Using the Automatically Created Workspace

To avoid manually creating the workspace and project, simply create your files and run them. To run those files, VC++ will automatically prompt you and create a workspace and project for you.

You can create a new file through the same dialogue box which you used in Section 1.2.1 to create your workspace. When that dialogue box appears select the *Files* tab. A list of names will appear below the tab indicating the various types of files you can create. Select *C++ Source File*.

¹ You have to follow the same steps if you wish to add header and resource files in their appropriate directories.
Introduction

Then type the name for your new .cpp file on the right hand side and indicate the appropriate directory where you would like to keep this file. Then press the OK button.

When you press OK, a blank file will be opened on the right-hand side of your screen. Now you can start typing your code in this file. A header file can be created in the same way. You will have to choose the C/C++ Header File option from the dialogue box shown above.

Regardless of which way you use to create your files, you will need to compile and build these files to be able to Run your program.

1.3 Creating a simple program

Let us assume that we have created a .cpp file (either through the steps explained in section 1.2.1 or through the ones explained in section 1.2.2)

Type the following piece of code in your file:
Now save your program from the *File* menu and press the *Compile* button (or *Ctrl+F7*) to compile your program. The location of the Compile icon has been indicated in the diagram above.

When you try to compile the program this way, a message will appear – indicating that an active project workspace does not exist. Press *Yes* so that a new default workspace is created. In this way, you will be by-passing some of the steps explained in section 1.2.1 earlier, by not manually creating a workspace and a project.

You will notice that when you press *Yes*, two new windows will appear on your screen; one on the left of your .cpp file and the other below your .cpp file.

The window on your left will tell you what workspace and project you are working on. The window at the bottom will give you the error reports when it compiles and builds your program.

If your program has been compiled successfully without any errors, you should see something similar to the following message in the bottom window on your screen:

```
------------------------Configuration: firstprogram - Win32 Debug------------------------
Compiling...
firstprogram.cpp

firstprogram.obj - 0 error(s), 0 warning(s)
```

In this example, *firstprogram.cpp* was my .cpp file and the compiler compiled it to create a *firstprogram.obj* file. This message also shows that while the compiler tried to create the .obj (Object) file, it encountered 0 error(s), 0 warnings(s). Had errors or warnings...
been encountered, this window would have contained some report/message regarding those errors and warnings.

Now build an executable file for your program by clicking the button to the right of the Compile button (or by pressing F7).

Again, in case of a successful building procedure, you will notice a configuration report in the bottom window of your screen:

--------------------Configuration: firstprogram - Win32 Debug---------------------
Linking...

firstprogram.exe - 0 error(s), 0 warning(s)

Now execute your program by pressing the button with a red exclamation mark (appearing on the second-right spot to the Build button) or by pressing Ctrl+F5.

This is what should appear on your screen: