Introduction
To
Microsoft Visual C++ 6.0

By: Shahed Shahir
Email: shahir@uwindsor.ca
Office hour: Mondays 9:00am-11:00am
Requirements

• Object Oriented Programming
• C++ programming
Introduction

Advantage:

Microsoft Visual C++ is a powerful tool. Visual C++ is inherited window’s functionality. In fact, developers can manipulate hardware and software at the same time; however, no other programming platforms can provide such a powerful tool for users in a windows operating system.

Microsoft Visual C++ includes Comprehensive debug tools.
Introduction

Disadvantage:

In order to monopolize computer software market, Microsoft does not reveal the key functions to public.

Reference

Microsoft Developer Network
http://msdn.microsoft.com/
Online Help
US$3000.00 for helping in a single project
US$240.00 for a single solution

My recommendation is to learn C++ and Object Oriented software Programming)
Computer Aided Design II (Professor M. Sid-Ahmed)
Outline

I. Console Application
II. Dialog Based Application
I. Console Application

• Start new project
1. Choose the project type as a Win32 console Application.

2. Type the project name as you wish.
What kind of Console Application do you want to create?

- An empty project.
- A simple application.
- A "Hello, World!" application.
- An application that supports MFC.
Win32 Console Application will create a new skeleton project with the following specifications:

+ Simple Win32 console application.
+ Prints "Hello, World!" to the console and then exits.

Main: HelloWorld.cpp
Precompiled Header: Stdafx.h and Stdafx.cpp

Project Directory:
C:\Documents and Settings\Shahed\My Documents\88-446\HelloWorld
// HelloWorld.cpp : Defines the entry point for the console
//
#include "stdafx.h"

int main(int argc, char* argv[])
{
    printf("Hello World!\n");
    return 0;
}
EXAMPLE ONE
II. Dialog Based Application

1. Start new project
2. Choose the project type and name
Start new project
Microsoft Fundation Class Project

• Choose Application Wizard for MFC
• Type the desire project name
• **CObject** is the principal base class for the MFC.
• **CCmdTarget** is the base class for all the application architecture classes in MFC. All classes that handle messages are derived, directly or indirectly, from this class.
• The **CWnd** class provides the base functionality of all window classes in MFC for Window programming.
• The **CDialog** class is the base class used for displaying dialog boxes on the screen. (modal and modeless)
What type of application would you like to create?

- Single document
- Multiple documents
- Dialog based

Document/View architecture support?

What language would you like your resources in?

English [United States] [APPWZENU.DLL]
New Project Information

AppWizard will create a new skeleton project with the following specifications:

Application type of SimpleCalculator:
   Dialog-Based Application targeting:
      Win32

Classes to be created:
   Application: CSimpleCalculatorApp in SimpleCalculator.h and
   SimpleCalculator.cpp
   Dialog: CSimpleCalculatorDlg in SimpleCalculatorDlg.h and
   SimpleCalculatorDlg.cpp

Features:
   + About box on system menu
   + 3D Controls
   + Uses shared DLL implementation (MFC42.DLL)
   + ActiveX Controls support enabled
   + Localizable text in:
      English [United States]

Project Directory:
C:\Documents and Settings\Shahed\My
Documents\aaa_May2003\SimpleCalculator

[OK] [Cancel]
TODO: Place dialog controls here.
Functions

1. Switch
2. UpdateData
3. UpdateWindow
1. Switch

- The **switch** and **case** keywords evaluate *expression* and execute any statement associated the initial expression.

  ```
  switch( expression )
  {
    [case constant-expression:]
    ...
    [statement]
    ...
    [default:
      statement]
  }
  ```
2. UpdateData

- **UpdateData** function, which is a member function of **CWnd**, initializes and retrieves data from a dialog box class.

**Syntax:**

- **BOOL UpdateData( BOOL Flag )**
  
  - **Return Value** is non zero if the operation is successful; otherwise 0.

- To initialize the Dialog box or any component on a dialog box, set input argument (**Flag**) to **FALSE**.
- To retrieve Data from the Dialog box or the components on a dialog box, set input argument (**Flag**) to **TRUE**.
3. UpdateWindow

- **UpdateWindow**, which is a member function of **CWnd**, updates the client area of the active window.

- **syntax:**
  - `void UpdateWindow();`
EXAMPLE TWO