**WINDOWS API**

**Introduction**

When writing applications for a PC running Microsoft Windows we need to use the Windows API (sometimes referred to as the Windows 32 API or Win32). When writing games we use it as a base for our DirectX applications. We also may want to use it to create game tools, like level editors, effects editors, resource managers etc.So even if you intend to work only with consoles it is still useful to learn to program. DirectX of course runs under Windows and some knowledge of the API is needed in order to get a DirectX application up and running.

The Windows API provides a lot of functionality that we will not be using. For our minimum needs we need to be able to create a window and receive user input. We may also want some dialogs and some menus the user can interface with.

The Windows API, informally WinAPI, is Microsoft's core set of application programming interfaces (APIs) available in the Microsoft Windows operating systems. It was formerly called the Win32 API; however, the name Windows API more accurately reflects its roots in 16-bit Windows and its support on 64-bit Windows. Almost all Windows programs interact with the Windows API; a small number (such as programs started early in the Windows startup process) use the Native API.

The functionality provided by the Windows API can be grouped into eight categories:

**Base Services**

Provide access to the fundamental resources available to a Windows system. Included are things like file systems, devices, processes and threads, and error handling.

**Advanced Services**

Provide access to functionality that is an addition on the kernel. Included are things like the Windows registry, shutdown/restart the system (or abort), start/stop/create a Windows service, manage user accounts.

**Graphics Device Interface**

Provides functionality for outputting graphical content to monitors, printers and other output devices.Kernel-mode GDI support is provided by win32k.sys which communicates directly with the graphics driver.

**User Interface**

Provides the functionality to create and manage screen windows and most basic controls, such as buttons and scrollbars, receive mouse and keyboard input, and other functionality associated with the GUI part of Windows.

**Common Dialog Box Library**

Provides applications the standard dialog boxes for opening and saving files, choosing color and font etc.It is grouped under the *User Interface* category of the API.

**Common Control Library**

Gives applications access to some advanced controls provided by the operating system. These include things like status bars, progress bars, toolbars and tabs

**Windows Shell**

Component of the Windows API allows applications to access the functionality provided by the operating system shell, as well as change and enhance it. It is grouped under the User Interface category of the API.

**Network Services**

Give access to the various networking capabilities of the operating system. Its sub-components include NetBIOS, Winsock, NetDDE, RPC and many others.

**Program Interaction**

The Windows API mostly concerns itself with the interaction between the operating system and an application. For communication between the different Windows applications among themselves, Microsoft has developed a series of technologies alongside the main Windows API. This started out with Dynamic Data Exchange (DDE), which was superseded by Object Linking and Embedding (OLE) and later by the Component Object Model (COM), Automation Objects, ActiveX controls, and the .NET Framework. There is not always a clear distinction between these technologies, and there is quite a lot of overlap.

The variety of terms is basically the result of grouping software mechanisms that relate to a particular aspect of software development. Automation specifically relates to exporting the functionality of an application or component (as an API) so that it can be controlled by another application instead of just by a human user, .NET is a self-contained general methodology and technology for developing Desktop and Web applications written in a variety of "Just in Time" compiled languages.

This program uses the basic set of classes that encapsulate the Windows API.

* Controller -- The bridge between Window Procedure and Object Oriented world.
* View -- Encapsulates the output of a Windows program.
* Canvas -- Encapsulated various Device Contexts and things you can do with them.
* Model -- The worker and the brain of your program. Doesn't deal with Windows at all.

**Relationship between API’s and MFC classes**

MFC supplies class CWnd to encapsulate the HWND handle of a window. The CWnd object is a C++ window object, distinct from the HWND that represents a Windows window but containing it. Use CWnd to derive your own child window classes, or use one of the many MFC classes derived from CWnd. Class CWnd is the base class for all windows, including frame windows, dialog boxes, child windows, controls, and control bars such as toolbars. A good understanding of the relationship between a C++ window object and an HWND is crucial for effective programming with MFC.

MFC provides some default functionality and management of windows, but you can derive your own class from CWnd and use its member functions to customize the provided functionality. You can create child windows by constructing a CWnd object and calling its Create member function, then customize the child windows using CWnd member functions. You can embed objects derived from CView, such as form views or tree views, in a frame window.

CWnd and its derived window classes provide constructors, destructors, and member functions to initialize the object, create the underlying Windows structures, and access the encapsulated HWND. CWnd also provides member functions that encapsulate Windows APIs for sending messages, accessing the window's state, converting coordinates, updating, scrolling, accessing the Clipboard, and many other tasks. Most Windows window-management APIs that take an HWND argument are encapsulated as member functions of CWnd. The names of the functions and their parameters are preserved in the CWnd member function. For details about the Windows APIs encapsulated by CWnd, see class CWnd.

**DOCUMENT-VIEW ARCHITECTURE**

The MFC document/view architecture makes it easy to support multiple views, multiple document types, splitter windows, and other valuable user-interface features.

The parts of the MFC framework most visible both to the user and to you, the programmer, are the document and view. Most of your work in developing an application with the framework goes into writing your document and view classes. This article family describes:

* The purposes of documents and views and how they interact in the framework.
* What you must do to implement them.

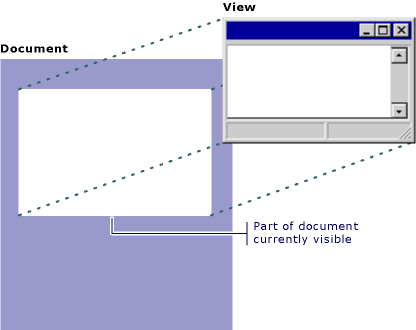
At the heart of document/view are four key classes:

The CDocument (or COleDocument) class: It supports objects used to store or control your program's data and provides the basic functionality for programmer-defined document classes. A document represents the unit of data that the user typically opens with the Open command on the File menu and saves with the Save command on the File menu.

The CView (or one of its many derived classes): It provides the basic functionality for programmer-defined view classes. A view is attached to a document and acts as an intermediary between the document and the user: the view renders an image of the document on the screen and interprets user input as operations upon the document. The view also renders the image for both printing and print preview.

CFrameWnd(or one of its variations): It supports objects that provides the frame around one or more views of a document.

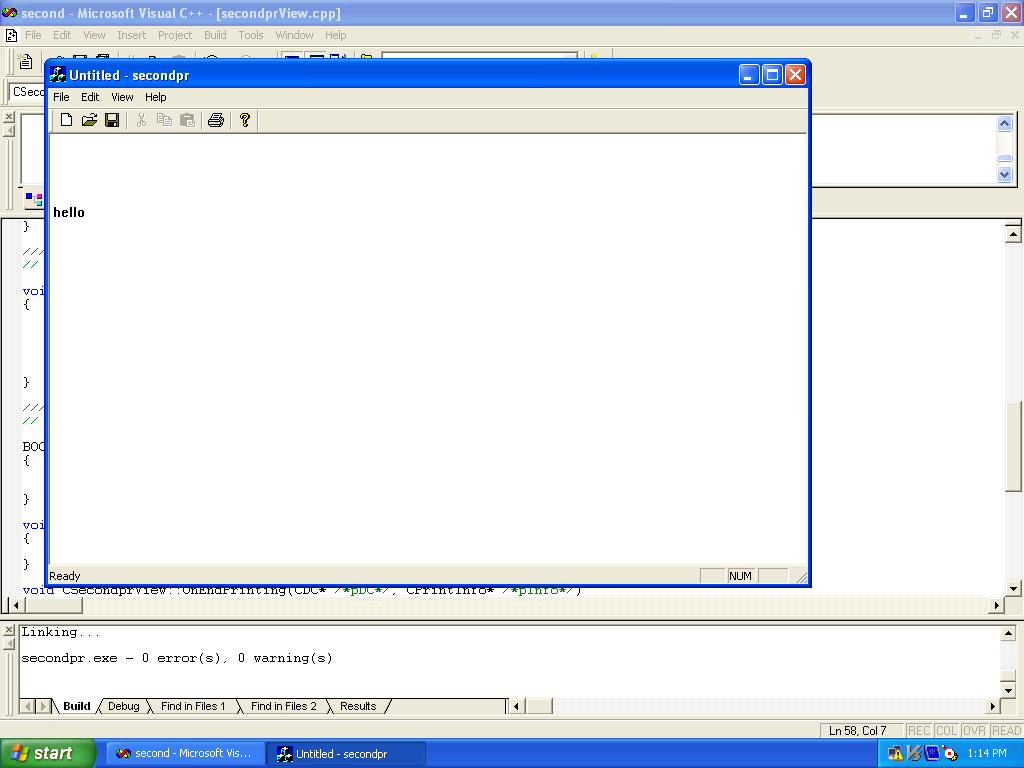
CDocTemplate(or CSingleDocTemplateor CMultiDocTemplate): It supports an object that coordinates one or more existing documents of a given type and manages creating the correct document, view, and frame window objects for that type.



The adjacent figure shows the relationship between a document and its view.

The document/view implementation in the class library separates the data itself from its display and from user operations on the data. All changes to the data are managed through the document class. The view calls this interface to access and update the data.

Documents, their associated views, and the frame windows that frame the views are created by a document template. The document template is responsible for creating and managing all documents of one document type.



**Write console programming in VC++ for two functions-**

1. **Take data**
2. **Display data**

# include<iostream>;

#include<conio.h>;

using namespace std;

class abc

{

public:

int r;

char n[7];

void getdata();

void displaydata();

};

void abc::getdata()

{

cout<<" enter name";

cin>>n;

cout<<" enter roll no";

cin>>r;

}

void abc::displaydata()

{

cout<<" my name is"<<n;

cout<<"my roll no is"<<r;

}

void main()

{

abc b;

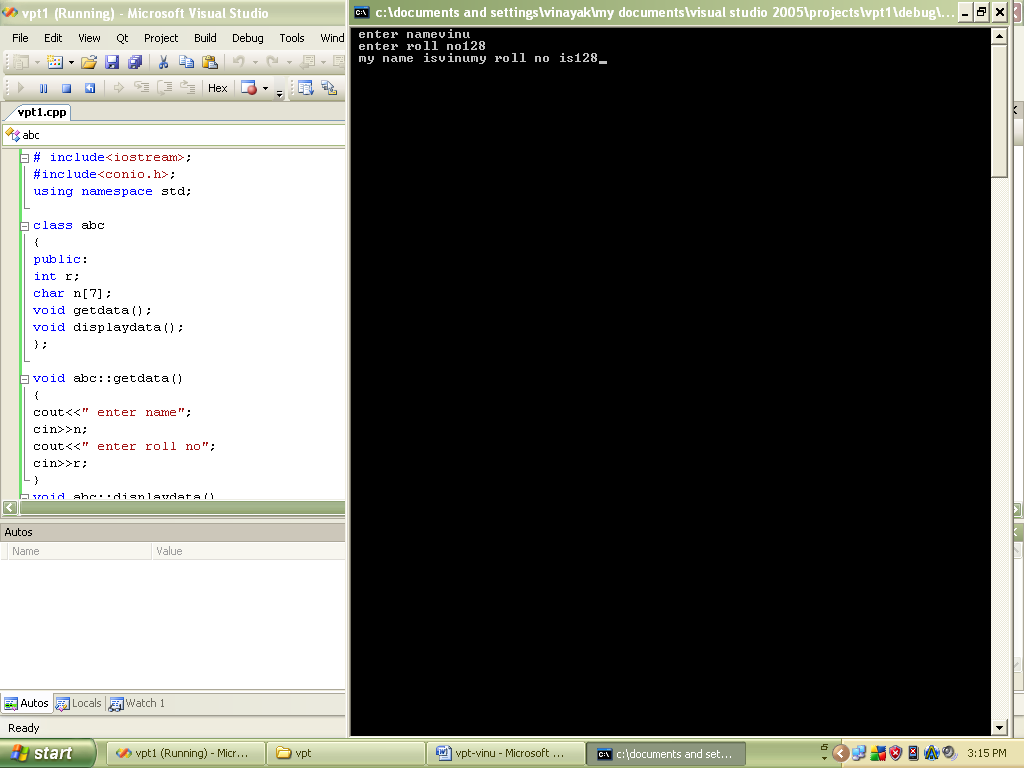
b.getdata();

b.displaydata();

getch();

}

**OUTPUT:**



**Write console programming in VC++ for inheritance**

#include<iostream>;

#include<conio.h>;

using namespace std;

class BASE

{

int a;

public:

int b;

void getdata()

{

cin>>a;

}

void putdata()

{

cout<<a;

}

};

class Derived : public BASE

{

int p;

public:

int q;

};

void main()

{

cout<<"SIZE OF THE BASE CLASS: "<<sizeof(BASE)<<"\n";

cout<<"SIZE OF THE DERIVED CLASS: "<<sizeof(Derived)<<"\n";

BASE obj1;

Derived obj2;

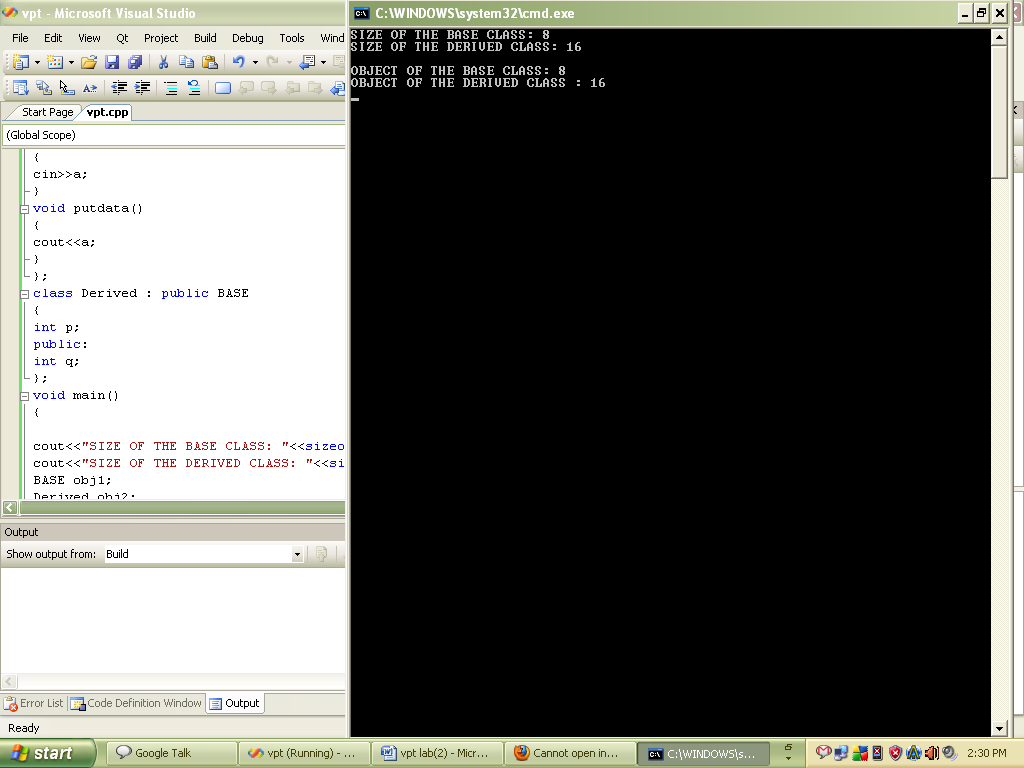
cout<<endl<<"OBJECT OF THE BASE CLASS: "<<sizeof(obj1)<<"\n";

cout<<"OBJECT OF THE DERIVED CLASS : "<<sizeof(obj2)<<"\n";

getch();

}

**OUTPUT:**



**Write console programming in VC++ for function overloading**

#include<iostream>;

#include<conio.h>;

using namespace std;

int area(float radius);

int area(int lenght,int breadth);

void main()

{

int ch,a,b,ans;

cout<<"ENTER YOUR CHOICE:";

cin>>ch;

switch (ch)

{

case 1:

cout<<"ENTER THE RADIUS OF THE CIRCLE:";

cin>>a;

ans=area(a);

cout<<"ANSWERS: "<<ans;

break;

case 2:

cout<<"ENTER THE LENGHT OF THE RECTANGLE:";

cin>>a;

cout<<"ENTER THE BREADTH OF THE RECTANGLE:";

cin>>b;

ans=area(a,b);

cout<<"ANSWERS:"<<ans;

break;

default:

cout<<"WRONG CHOICE:";

break;

};

getch();

}

int area(float radius)

{

int area=radius\*3.14\*radius;

return(area);

}

int area(int lenght,int breadth)

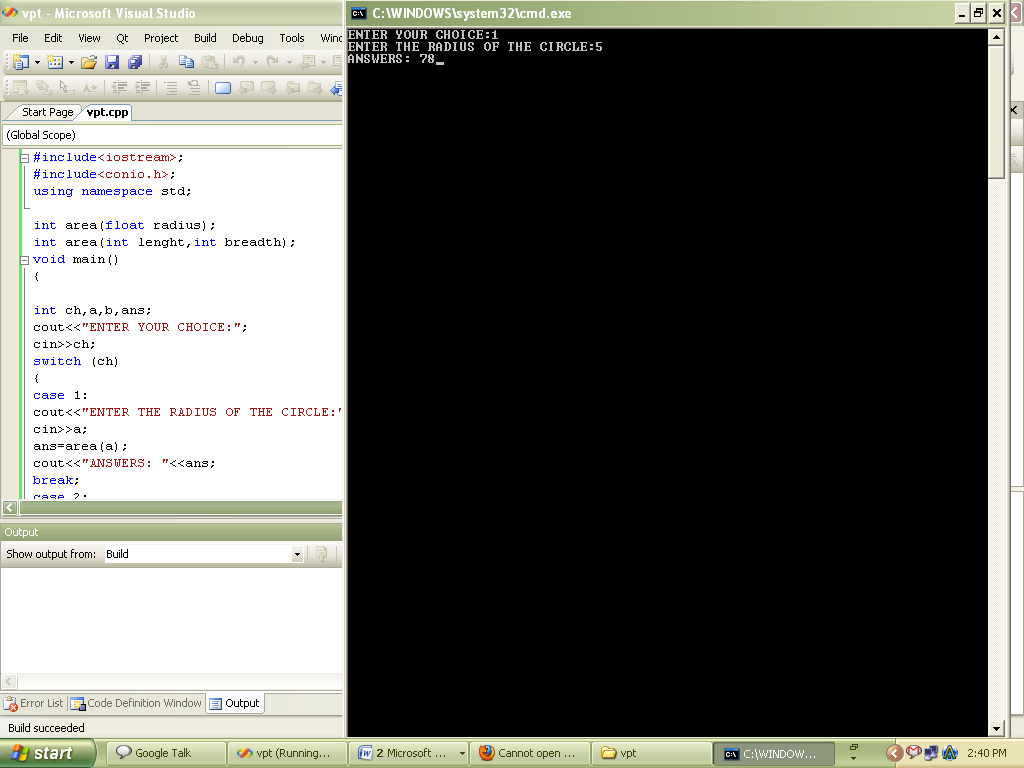
{

int area=lenght\*breadth;

return(area);

}

**OUTPUT:**



**Windows Programming in VC++ to print Welcome Messages**

**Welcomemsg.h and Welcomemsg.cpp**

// Welcomemsg.h : main header file for the WELCOMEMSG application

//

#if !defined(AFX\_WELCOMEMSG\_H\_\_65CAD26C\_1D91\_47C3\_83E6\_72C09ED9523D\_\_INCLUDED\_)

#define AFX\_WELCOMEMSG\_H\_\_65CAD26C\_1D91\_47C3\_83E6\_72C09ED9523D\_\_INCLUDED\_

#if \_MSC\_VER > 1000

#pragma once

#endif // \_MSC\_VER > 1000

#ifndef \_\_AFXWIN\_H\_\_

#error include 'stdafx.h' before including this file for PCH

#endif

#include "resource.h" // main symbols

/////////////////////////////////////////////////////////////////////////////

// CWelcomemsgApp:

// See Welcomemsg.cpp for the implementation of this class

//

class CWelcomemsgApp : public CWinApp

{

public:

CWelcomemsgApp();

// Overrides

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(CWelcomemsgApp)

public:

virtual BOOL InitInstance();

//}}AFX\_VIRTUAL

// Implementation

//{{AFX\_MSG(CWelcomemsgApp)

afx\_msg void OnAppAbout();

// NOTE - the ClassWizard will add and remove member functions here.

// DO NOT EDIT what you see in these blocks of generated code !

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

};

/////////////////////////////////////////////////////////////////////////////

//{{AFX\_INSERT\_LOCATION}}

// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX\_WELCOMEMSG\_H\_\_65CAD26C\_1D91\_47C3\_83E6\_72C09ED9523D\_\_INCLUDED\_)

// Welcomemsg.cpp : Defines the class behaviors for the application.

//

#include "stdafx.h"

#include "Welcomemsg.h"

#include "MainFrm.h"

#include "WelcomemsgDoc.h"

#include "WelcomemsgView.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

/////////////////////////////////////////////////////////////////////////////

// CWelcomemsgApp

BEGIN\_MESSAGE\_MAP(CWelcomemsgApp, CWinApp)

//{{AFX\_MSG\_MAP(CWelcomemsgApp)

ON\_COMMAND(ID\_APP\_ABOUT, OnAppAbout)

// NOTE - the ClassWizard will add and remove mapping macros here.

// DO NOT EDIT what you see in these blocks of generated code!

//}}AFX\_MSG\_MAP

// Standard file based document commands

ON\_COMMAND(ID\_FILE\_NEW, CWinApp::OnFileNew)

ON\_COMMAND(ID\_FILE\_OPEN, CWinApp::OnFileOpen)

// Standard print setup command

ON\_COMMAND(ID\_FILE\_PRINT\_SETUP, CWinApp::OnFilePrintSetup)

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// CWelcomemsgApp construction

CWelcomemsgApp::CWelcomemsgApp()

{

// TODO: add construction code here,

// Place all significant initialization in InitInstance

}

/////////////////////////////////////////////////////////////////////////////

// The one and only CWelcomemsgApp object

CWelcomemsgApp theApp;

/////////////////////////////////////////////////////////////////////////////

// CWelcomemsgApp initialization

BOOL CWelcomemsgApp::InitInstance()

{

AfxEnableControlContainer();

// Standard initialization

// If you are not using these features and wish to reduce the size

// of your final executable, you should remove from the following

// the specific initialization routines you do not need.

#ifdef \_AFXDLL

Enable3dControls(); // Call this when using MFC in a shared DLL

#else

Enable3dControlsStatic(); // Call this when linking to MFC statically

#endif

// Change the registry key under which our settings are stored.

// TODO: You should modify this string to be something appropriate

// such as the name of your company or organization.

SetRegistryKey(\_T("Local AppWizard-Generated Applications"));

LoadStdProfileSettings(); // Load standard INI file options (including MRU)

// Register the application's document templates. Document templates

// serve as the connection between documents, frame windows and views.

CSingleDocTemplate\* pDocTemplate;

pDocTemplate = new CSingleDocTemplate(

IDR\_MAINFRAME,

RUNTIME\_CLASS(CWelcomemsgDoc),

RUNTIME\_CLASS(CMainFrame), // main SDI frame window

RUNTIME\_CLASS(CWelcomemsgView));

AddDocTemplate(pDocTemplate);

// Parse command line for standard shell commands, DDE, file open

CCommandLineInfo cmdInfo;

ParseCommandLine(cmdInfo);

// Dispatch commands specified on the command line

if (!ProcessShellCommand(cmdInfo))

return FALSE;

// The one and only window has been initialized, so show and update it.

m\_pMainWnd->ShowWindow(SW\_SHOW);

m\_pMainWnd->UpdateWindow();

return TRUE;

}

/////////////////////////////////////////////////////////////////////////////

// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog

{

public:

CAboutDlg();

// Dialog Data

//{{AFX\_DATA(CAboutDlg)

enum { IDD = IDD\_ABOUTBOX };

//}}AFX\_DATA

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(CAboutDlg)

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV support

//}}AFX\_VIRTUAL

// Implementation

protected:

//{{AFX\_MSG(CAboutDlg)

// No message handlers

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)

{

//{{AFX\_DATA\_INIT(CAboutDlg)

//}}AFX\_DATA\_INIT

}

void CAboutDlg::DoDataExchange(CDataExchange\* pDX)

{

CDialog::DoDataExchange(pDX);

//{{AFX\_DATA\_MAP(CAboutDlg)

//}}AFX\_DATA\_MAP

}

BEGIN\_MESSAGE\_MAP(CAboutDlg, CDialog)

//{{AFX\_MSG\_MAP(CAboutDlg)

// No message handlers

//}}AFX\_MSG\_MAP

END\_MESSAGE\_MAP()

// App command to run the dialog

void CWelcomemsgApp::OnAppAbout()

{

CAboutDlg aboutDlg;

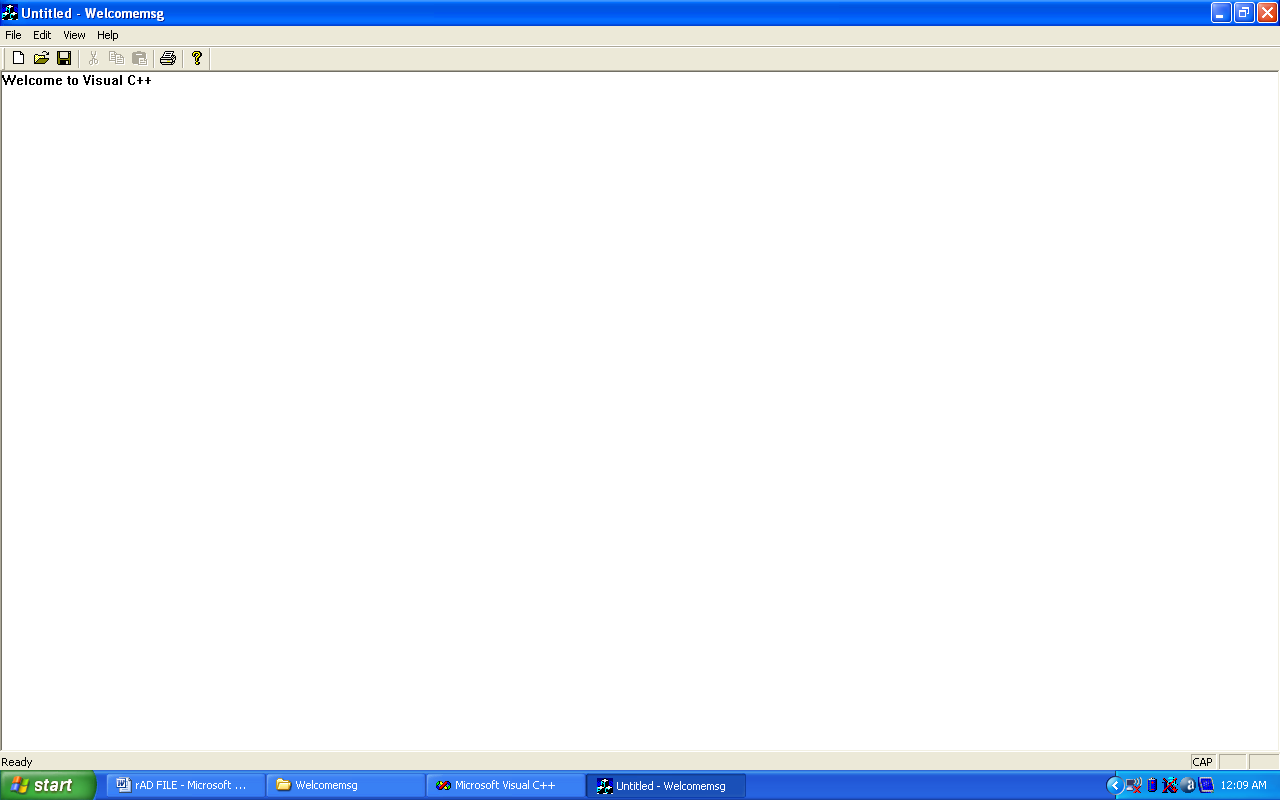
aboutDlg.DoModal();

}

/////////////////////////////////////////////////////////////////////////////

// CWelcomemsgApp message handlers

**OUTPUT:**



**Write a Program for Accepting Keystrokes from the Keyboard In VC++.**

KeyDoc.h and KeyDoc.cpp

// KeyDoc.h : interface of the CKeyDoc class

//

/////////////////////////////////////////////////////////////////////////////

#if !defined(AFX\_KEYDOC\_H\_\_33E9B9D2\_67BA\_4A34\_9679\_92E42DD6324F\_\_INCLUDED\_)

#define AFX\_KEYDOC\_H\_\_33E9B9D2\_67BA\_4A34\_9679\_92E42DD6324F\_\_INCLUDED\_

#if \_MSC\_VER > 1000

#pragma once

#endif // \_MSC\_VER > 1000

class CKeyDoc : public CDocument

{

protected: // create from serialization only

CKeyDoc();

DECLARE\_DYNCREATE(CKeyDoc)

CString xString;

// Attributes

public:

// Operations

public:

// Overrides

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(CKeyDoc)

public:

virtual BOOL OnNewDocument();

virtual void Serialize(CArchive& ar);

//}}AFX\_VIRTUAL

// Implementation

public:

virtual ~CKeyDoc();

#ifdef \_DEBUG

virtual void AssertValid() const;

virtual void Dump(CDumpContext& dc) const;

#endif

protected:

// Generated message map functions

protected:

//{{AFX\_MSG(CKeyDoc)

// NOTE - the ClassWizard will add and remove member functions here.

// DO NOT EDIT what you see in these blocks of generated code !

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

};

/////////////////////////////////////////////////////////////////////////////

//{{AFX\_INSERT\_LOCATION}}

// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX\_KEYDOC\_H\_\_33E9B9D2\_67BA\_4A34\_9679\_92E42DD6324F\_\_INCLUDED\_)

// KeyDoc.cpp : implementation of the CKeyDoc class

//

#include "stdafx.h"

#include "Key.h"

#include "KeyDoc.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

/////////////////////////////////////////////////////////////////////////////

// CKeyDoc

IMPLEMENT\_DYNCREATE(CKeyDoc, CDocument)

BEGIN\_MESSAGE\_MAP(CKeyDoc, CDocument)

//{{AFX\_MSG\_MAP(CKeyDoc)

// NOTE - the ClassWizard will add and remove mapping macros here.

// DO NOT EDIT what you see in these blocks of generated code!

//}}AFX\_MSG\_MAP

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// CKeyDoc construction/destruction

CKeyDoc::CKeyDoc()

{

// TODO: add one-time construction code here

xString=" ";

}

CKeyDoc::~CKeyDoc()

{

}

BOOL CKeyDoc::OnNewDocument()

{

if (!CDocument::OnNewDocument())

return FALSE;

// TODO: add reinitialization code here

// (SDI documents will reuse this document)

return TRUE;

}

/////////////////////////////////////////////////////////////////////////////

// CKeyDoc serialization

void CKeyDoc::Serialize(CArchive& ar)

{

if (ar.IsStoring())

{

// TODO: add storing code here

}

else

{

// TODO: add loading code here

}

}

/////////////////////////////////////////////////////////////////////////////

// CKeyDoc diagnostics

#ifdef \_DEBUG

void CKeyDoc::AssertValid() const

{

CDocument::AssertValid();

}

void CKeyDoc::Dump(CDumpContext& dc) const

{

CDocument::Dump(dc);

}

#endif //\_DEBUG

/////////////////////////////////////////////////////////////////////////////

// CKeyDoc commands

KeyView.h and KeyView.cpp

// KeyView.h : interface of the CKeyView class

//

/////////////////////////////////////////////////////////////////////////////

#if !defined(AFX\_KEYVIEW\_H\_\_0B6BAE47\_5F2C\_4C35\_8F22\_D27717824792\_\_INCLUDED\_)

#define AFX\_KEYVIEW\_H\_\_0B6BAE47\_5F2C\_4C35\_8F22\_D27717824792\_\_INCLUDED\_

#if \_MSC\_VER > 1000

#pragma once

#endif // \_MSC\_VER > 1000

class CKeyView : public CView

{

protected: // create from serialization only

CKeyView();

DECLARE\_DYNCREATE(CKeyView)

// Attributes

public:

CKeyDoc\* GetDocument();

// Operations

public:

// Overrides

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(CKeyView)

public:

virtual void OnDraw(CDC\* pDC); // overridden to draw this view

virtual BOOL PreCreateWindow(CREATESTRUCT& cs);

protected:

virtual BOOL OnPreparePrinting(CPrintInfo\* pInfo);

virtual void OnBeginPrinting(CDC\* pDC, CPrintInfo\* pInfo);

virtual void OnEndPrinting(CDC\* pDC, CPrintInfo\* pInfo);

//}}AFX\_VIRTUAL

// Implementation

public:

virtual ~CKeyView();

#ifdef \_DEBUG

virtual void AssertValid() const;

virtual void Dump(CDumpContext& dc) const;

#endif

protected:

// Generated message map functions

protected:

//{{AFX\_MSG(CKeyView)

afx\_msg void OnChar(UINT nChar, UINT nRepCnt, UINT nFlags);

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

};

#ifndef \_DEBUG // debug version in KeyView.cpp

inline CKeyDoc\* CKeyView::GetDocument()

{ return (CKeyDoc\*)m\_pDocument; }

#endif

/////////////////////////////////////////////////////////////////////////////

//{{AFX\_INSERT\_LOCATION}}

// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX\_KEYVIEW\_H\_\_0B6BAE47\_5F2C\_4C35\_8F22\_D27717824792\_\_INCLUDED\_) // KeyView.cpp : implementation of the CKeyView class

//

#include "stdafx.h"

#include "Key.h"

#include "KeyDoc.h"

#include "KeyView.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

/////////////////////////////////////////////////////////////////////////////

// CKeyView

IMPLEMENT\_DYNCREATE(CKeyView, CView)

BEGIN\_MESSAGE\_MAP(CKeyView, CView)

//{{AFX\_MSG\_MAP(CKeyView)

ON\_WM\_CHAR()

//}}AFX\_MSG\_MAP

// Standard printing commands

ON\_COMMAND(ID\_FILE\_PRINT, CView::OnFilePrint)

ON\_COMMAND(ID\_FILE\_PRINT\_DIRECT, CView::OnFilePrint)

ON\_COMMAND(ID\_FILE\_PRINT\_PREVIEW, CView::OnFilePrintPreview)

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// CKeyView construction/destruction

CKeyView::CKeyView()

{

// TODO: add construction code here }

CKeyView::~CKeyView()

{

}

BOOL CKeyView::PreCreateWindow(CREATESTRUCT& cs)

{

// TODO: Modify the Window class or styles here by modifying

// the CREATESTRUCT cs

return CView::PreCreateWindow(cs);

}

/////////////////////////////////////////////////////////////////////////////

// CKeyView drawing

void CKeyView::OnDraw(CDC\* pDC)

{

CKeyDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

pDC->TextOut(0,0,pDoc->xString);

// TODO: add draw code for native data here

}

/////////////////////////////////////////////////////////////////////////////

// CKeyView printing

BOOL CKeyView::OnPreparePrinting(CPrintInfo\* pInfo)

{

// default preparation

return DoPreparePrinting(pInfo);

}

void CKeyView::OnBeginPrinting(CDC\* /\*pDC\*/, CPrintInfo\* /\*pInfo\*/)

{

// TODO: add extra initialization before printing

}

void CKeyView::OnEndPrinting(CDC\* /\*pDC\*/, CPrintInfo\* /\*pInfo\*/)

{

// TODO: add cleanup after printing

}

/////////////////////////////////////////////////////////////////////////////

// CKeyView diagnostics

#ifdef \_DEBUG

void CKeyView::AssertValid() const

{

CView::AssertValid();

}

void CKeyView::Dump(CDumpContext& dc) const

{

CView::Dump(dc);

}

CKeyDoc\* CKeyView::GetDocument() // non-debug version is inline

{

ASSERT(m\_pDocument->IsKindOf(RUNTIME\_CLASS(CKeyDoc)));

return (CKeyDoc\*)m\_pDocument;

}

#endif //\_DEBUG

/////////////////////////////////////////////////////////////////////////////

// CKeyView message handlers

void CKeyView::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)

{

// TODO: Add your message handler code here and/or call default

CKeyDoc\* pDoc =GetDocument();

ASSERT\_VALID(pDoc);

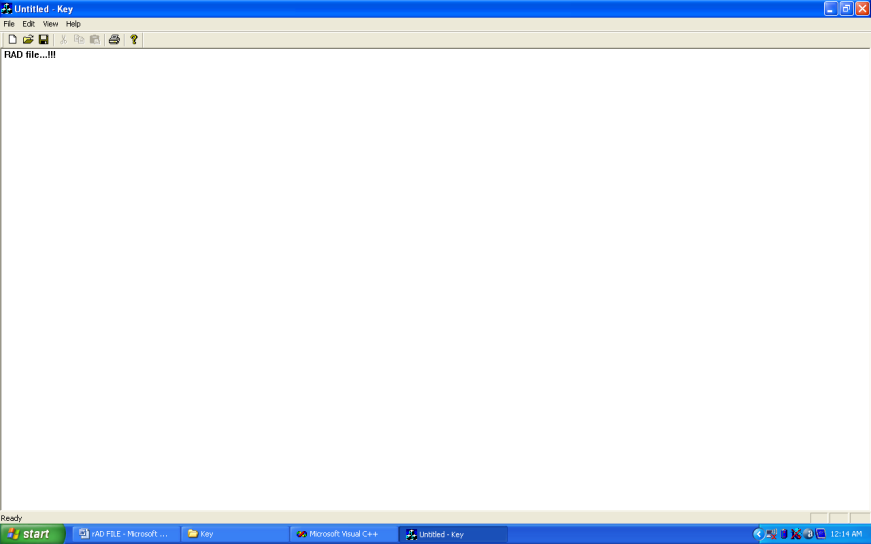
pDoc->xString+=nChar;

Invalidate();

CView::OnChar(nChar, nRepCnt, nFlags);

}

**OUTPUT:**



**Write a Program To Add Menu Items To a Menu in VC++.**

MenusDoc.h and MenusDoc.cpp

// MenusDoc.h : interface of the CMenusDoc class

//

/////////////////////////////////////////////////////////////////////////////

#if !defined(AFX\_MENUSDOC\_H\_\_ED4689EF\_FD8C\_4365\_8B35\_423527658D45\_\_INCLUDED\_)

#define AFX\_MENUSDOC\_H\_\_ED4689EF\_FD8C\_4365\_8B35\_423527658D45\_\_INCLUDED\_

#if \_MSC\_VER > 1000

#pragma once

#endif // \_MSC\_VER > 1000

class CMenusDoc : public CDocument

{

protected: // create from serialization only

CMenusDoc();

DECLARE\_DYNCREATE(CMenusDoc)

// Attributes

public:

// Operations

public:

// Overrides

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(CMenusDoc)

public:

virtual BOOL OnNewDocument();

virtual void Serialize(CArchive& ar);

//}}AFX\_VIRTUAL

// Implementation

public:

virtual ~CMenusDoc();

CString StringData;

#ifdef \_DEBUG

virtual void AssertValid() const;

virtual void Dump(CDumpContext& dc) const;

#endif

protected:

// Generated message map functions

protected:

//{{AFX\_MSG(CMenusDoc)

// NOTE - the ClassWizard will add and remove member functions here.

// DO NOT EDIT what you see in these blocks of generated code !

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

};

/////////////////////////////////////////////////////////////////////////////

//{{AFX\_INSERT\_LOCATION}}

// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX\_MENUSDOC\_H\_\_ED4689EF\_FD8C\_4365\_8B35\_423527658D45\_\_INCLUDED\_)

// MenusDoc.cpp : implementation of the CMenusDoc class

//

#include "stdafx.h"

#include "Menus.h"

#include "MenusDoc.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

/////////////////////////////////////////////////////////////////////////////

// CMenusDoc

IMPLEMENT\_DYNCREATE(CMenusDoc, CDocument)

BEGIN\_MESSAGE\_MAP(CMenusDoc, CDocument)

//{{AFX\_MSG\_MAP(CMenusDoc)

// NOTE - the ClassWizard will add and remove mapping macros here.

// DO NOT EDIT what you see in these blocks of generated code!

//}}AFX\_MSG\_MAP

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// CMenusDoc construction/destruction

CMenusDoc::CMenusDoc()

{

// TODO: add one-time construction code here

StringData="";

}

CMenusDoc::~CMenusDoc()

{

}

BOOL CMenusDoc::OnNewDocument()

{

if (!CDocument::OnNewDocument())

return FALSE;

// TODO: add reinitialization code here

// (SDI documents will reuse this document)

return TRUE;

}

/////////////////////////////////////////////////////////////////////////////

// CMenusDoc serialization

void CMenusDoc::Serialize(CArchive& ar)

{

if (ar.IsStoring())

{

// TODO: add storing code here

}

else

{

// TODO: add loading code here

}

}

/////////////////////////////////////////////////////////////////////////////

// CMenusDoc diagnostics

#ifdef \_DEBUG

void CMenusDoc::AssertValid() const

{

CDocument::AssertValid();

}

void CMenusDoc::Dump(CDumpContext& dc) const

{

CDocument::Dump(dc);

}

#endif //\_DEBUG

/////////////////////////////////////////////////////////////////////////////

// CMenusDoc commands

MenusView.h and MenusView.cpp

// MenusView.h : interface of the CMenusView class

//

/////////////////////////////////////////////////////////////////////////////

#if !defined(AFX\_MENUSVIEW\_H\_\_77CB4679\_BA57\_4EA4\_840D\_B1E445EF5F43\_\_INCLUDED\_)

#define AFX\_MENUSVIEW\_H\_\_77CB4679\_BA57\_4EA4\_840D\_B1E445EF5F43\_\_INCLUDED\_

#if \_MSC\_VER > 1000

#pragma once

#endif // \_MSC\_VER > 1000

class CMenusView : public CView

{

protected: // create from serialization only

CMenusView();

DECLARE\_DYNCREATE(CMenusView)

// Attributes

public:

CMenusDoc\* GetDocument();

// Operations

public:

// Overrides

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(CMenusView)

public:

virtual void OnDraw(CDC\* pDC); // overridden to draw this view

virtual BOOL PreCreateWindow(CREATESTRUCT& cs);

protected:

virtual BOOL OnPreparePrinting(CPrintInfo\* pInfo);

virtual void OnBeginPrinting(CDC\* pDC, CPrintInfo\* pInfo);

virtual void OnEndPrinting(CDC\* pDC, CPrintInfo\* pInfo);

//}}AFX\_VIRTUAL

// Implementation

public:

virtual ~CMenusView();

#ifdef \_DEBUG

virtual void AssertValid() const;

virtual void Dump(CDumpContext& dc) const;

#endif

protected:

// Generated message map functions

protected:

//{{AFX\_MSG(CMenusView)

afx\_msg void OnFilePrintwelcome();

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

};

#ifndef \_DEBUG // debug version in MenusView.cpp

inline CMenusDoc\* CMenusView::GetDocument()

{ return (CMenusDoc\*)m\_pDocument; }

#endif

/////////////////////////////////////////////////////////////////////////////

//{{AFX\_INSERT\_LOCATION}}

// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX\_MENUSVIEW\_H\_\_77CB4679\_BA57\_4EA4\_840D\_B1E445EF5F43\_\_INCLUDED\_)

// MenusView.cpp : implementation of the CMenusView class

//

#include "stdafx.h"

#include "Menus.h"

#include "MenusDoc.h"

#include "MenusView.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

/////////////////////////////////////////////////////////////////////////////

// CMenusView

IMPLEMENT\_DYNCREATE(CMenusView, CView)

BEGIN\_MESSAGE\_MAP(CMenusView, CView)

//{{AFX\_MSG\_MAP(CMenusView)

ON\_COMMAND(ID\_FILE\_PRINTWELCOME, OnFilePrintwelcome)

//}}AFX\_MSG\_MAP

// Standard printing commands

ON\_COMMAND(ID\_FILE\_PRINT, CView::OnFilePrint)

ON\_COMMAND(ID\_FILE\_PRINT\_DIRECT, CView::OnFilePrint)

ON\_COMMAND(ID\_FILE\_PRINT\_PREVIEW, CView::OnFilePrintPreview)

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// CMenusView construction/destruction

CMenusView::CMenusView()

{

// TODO: add construction code here

}

CMenusView::~CMenusView()

{

}

BOOL CMenusView::PreCreateWindow(CREATESTRUCT& cs)

{

// TODO: Modify the Window class or styles here by modifying

// the CREATESTRUCT cs

return CView::PreCreateWindow(cs);

}

/////////////////////////////////////////////////////////////////////////////

// CMenusView drawing

void CMenusView::OnDraw(CDC\* pDC)

{

CMenusDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

pDC->TextOut(0,0,pDoc->StringData);

// TODO: add draw code for native data here

}

/////////////////////////////////////////////////////////////////////////////

// CMenusView printing

BOOL CMenusView::OnPreparePrinting(CPrintInfo\* pInfo)

{

// default preparation

return DoPreparePrinting(pInfo);

}

void CMenusView::OnBeginPrinting(CDC\* /\*pDC\*/, CPrintInfo\* /\*pInfo\*/)

{

// TODO: add extra initialization before printing }

void CMenusView::OnEndPrinting(CDC\* /\*pDC\*/, CPrintInfo\* /\*pInfo\*/)

{

// TODO: add cleanup after printing

}

/////////////////////////////////////////////////////////////////////////////

// CMenusView diagnostics

#ifdef \_DEBUG

void CMenusView::AssertValid() const

{

CView::AssertValid();

}

void CMenusView::Dump(CDumpContext& dc) const

{

CView::Dump(dc);

}

CMenusDoc\* CMenusView::GetDocument() // non-debug version is inline

{

ASSERT(m\_pDocument->IsKindOf(RUNTIME\_CLASS(CMenusDoc)));

return (CMenusDoc\*)m\_pDocument;

}

#endif //\_DEBUG

/////////////////////////////////////////////////////////////////////////////

// CMenusView message handlers

void CMenusView::OnFilePrintwelcome()

{

// TODO: Add your command handler code here

CMenusDoc\* pDoc = GetDocument();

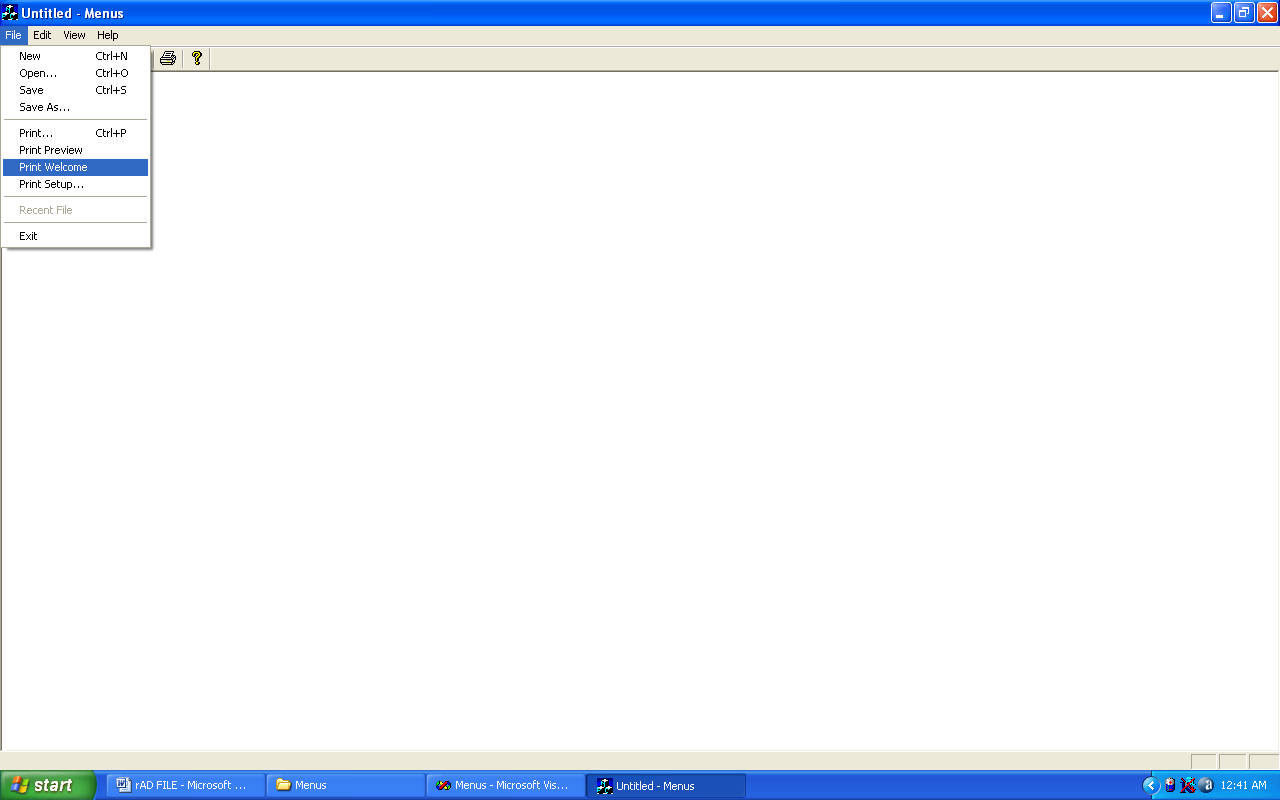
ASSERT\_VALID(pDoc);

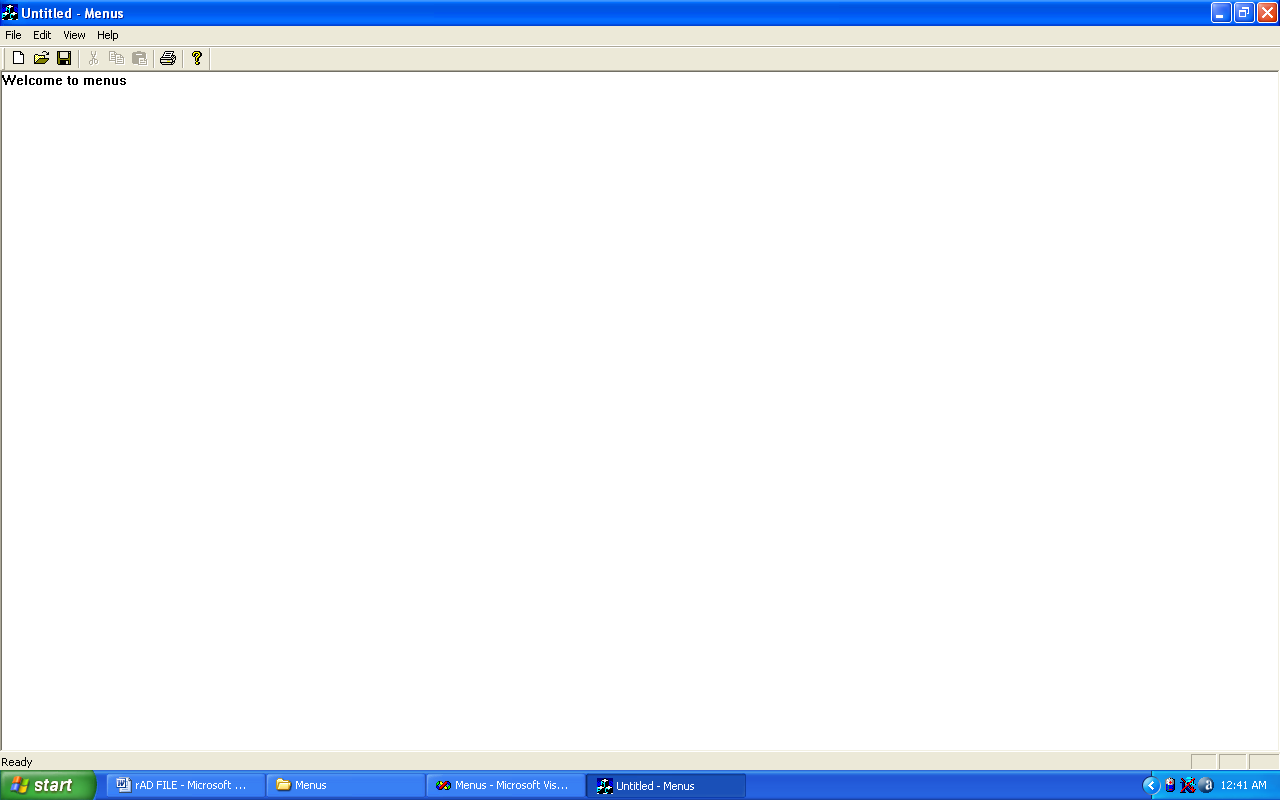
pDoc->StringData="Welcome to menus";

Invalidate();

}

**OUTPUT:**





**Write a Program To Create a Dialog Box In VC++.**

DialogsDoc.h and DialogsDoc.cpp

// DialogsDoc.h : interface of the CDialogsDoc class

//

/////////////////////////////////////////////////////////////////////////////

#if !defined(AFX\_DIALOGSDOC\_H\_\_869B6934\_7D9A\_4AA8\_AC36\_F4521438B018\_\_INCLUDED\_)

#define AFX\_DIALOGSDOC\_H\_\_869B6934\_7D9A\_4AA8\_AC36\_F4521438B018\_\_INCLUDED\_

#if \_MSC\_VER > 1000

#pragma once

#endif // \_MSC\_VER > 1000

class CDialogsDoc : public CDocument

{

protected: // create from serialization only

CDialogsDoc();

DECLARE\_DYNCREATE(CDialogsDoc)

// Attributes

public:

// Operations

public:

// Overrides

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(CDialogsDoc)

public:

virtual BOOL OnNewDocument();

virtual void Serialize(CArchive& ar);

//}}AFX\_VIRTUAL

// Implementation

public:

virtual ~CDialogsDoc();

CString StringData;

#ifdef \_DEBUG

virtual void AssertValid() const;

virtual void Dump(CDumpContext& dc) const;

#endif

protected:

// Generated message map functions

protected:

//{{AFX\_MSG(CDialogsDoc)

// NOTE - the ClassWizard will add and remove member functions here.

// DO NOT EDIT what you see in these blocks of generated code !

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

};

/////////////////////////////////////////////////////////////////////////////

//{{AFX\_INSERT\_LOCATION}}

// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX\_DIALOGSDOC\_H\_\_869B6934\_7D9A\_4AA8\_AC36\_F4521438B018\_\_INCLUDED\_)

// DialogsDoc.cpp : implementation of the CDialogsDoc class

//

#include "stdafx.h"

#include "Dialogs.h"

#include "DialogsDoc.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

/////////////////////////////////////////////////////////////////////////////

// CDialogsDoc

IMPLEMENT\_DYNCREATE(CDialogsDoc, CDocument)

BEGIN\_MESSAGE\_MAP(CDialogsDoc, CDocument)

//{{AFX\_MSG\_MAP(CDialogsDoc)

// NOTE - the ClassWizard will add and remove mapping macros here.

// DO NOT EDIT what you see in these blocks of generated code!

//}}AFX\_MSG\_MAP

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// CDialogsDoc construction/destruction

CDialogsDoc::CDialogsDoc()

{

// TODO: add one-time construction code here

StringData="";

}

CDialogsDoc::~CDialogsDoc()

{

}

BOOL CDialogsDoc::OnNewDocument()

{

if (!CDocument::OnNewDocument())

return FALSE;

// TODO: add reinitialization code here

// (SDI documents will reuse this document)

return TRUE;

}

/////////////////////////////////////////////////////////////////////////////

// CDialogsDoc serialization

void CDialogsDoc::Serialize(CArchive& ar)

{

if (ar.IsStoring())

{

// TODO: add storing code here

}

else

{

// TODO: add loading code here

}

}

/////////////////////////////////////////////////////////////////////////////

// CDialogsDoc diagnostics

#ifdef \_DEBUG

void CDialogsDoc::AssertValid() const

{

CDocument::AssertValid();

}

void CDialogsDoc::Dump(CDumpContext& dc) const

{

CDocument::Dump(dc);

}

#endif //\_DEBUG

/////////////////////////////////////////////////////////////////////////////

// CDialogsDoc commands

DialogsView.h and DialogsView.cpp

// DialogsView.h : interface of the CDialogsView class

//

/////////////////////////////////////////////////////////////////////////////

#if !defined(AFX\_DIALOGSVIEW\_H\_\_6F1A2E1B\_6A7F\_4CF5\_AD34\_89A2AD0D96D4\_\_INCLUDED\_)

#define AFX\_DIALOGSVIEW\_H\_\_6F1A2E1B\_6A7F\_4CF5\_AD34\_89A2AD0D96D4\_\_INCLUDED\_

#if \_MSC\_VER > 1000

#pragma once

#endif // \_MSC\_VER > 1000

class CDialogsView : public CView

{

protected: // create from serialization only

CDialogsView();

DECLARE\_DYNCREATE(CDialogsView)

// Attributes

public:

CDialogsDoc\* GetDocument();

// Operations

public:

// Overrides

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(CDialogsView)

public:

virtual void OnDraw(CDC\* pDC); // overridden to draw this view

virtual BOOL PreCreateWindow(CREATESTRUCT& cs);

protected:

virtual BOOL OnPreparePrinting(CPrintInfo\* pInfo);

virtual void OnBeginPrinting(CDC\* pDC, CPrintInfo\* pInfo);

virtual void OnEndPrinting(CDC\* pDC, CPrintInfo\* pInfo);

//}}AFX\_VIRTUAL

// Implementation

public:

virtual ~CDialogsView();

#ifdef \_DEBUG

virtual void AssertValid() const;

virtual void Dump(CDumpContext& dc) const;

#endif

protected:

// Generated message map functions

protected:

//{{AFX\_MSG(CDialogsView)

afx\_msg void OnFileShowdialog();

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

};

#ifndef \_DEBUG // debug version in DialogsView.cpp

inline CDialogsDoc\* CDialogsView::GetDocument()

{ return (CDialogsDoc\*)m\_pDocument; }

#endif

/////////////////////////////////////////////////////////////////////////////

//{{AFX\_INSERT\_LOCATION}}

// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX\_DIALOGSVIEW\_H\_\_6F1A2E1B\_6A7F\_4CF5\_AD34\_89A2AD0D96D4\_\_INCLUDED\_)

// DialogsView.cpp : implementation of the CDialogsView class

//

#include "stdafx.h"

#include "Dialogs.h"

#include "Dlg.h"

#include "DialogsDoc.h"

#include "DialogsView.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

/////////////////////////////////////////////////////////////////////////////

// CDialogsView

IMPLEMENT\_DYNCREATE(CDialogsView, CView)

BEGIN\_MESSAGE\_MAP(CDialogsView, CView)

//{{AFX\_MSG\_MAP(CDialogsView)

ON\_COMMAND(ID\_FILE\_SHOWDIALOG, OnFileShowdialog)

//}}AFX\_MSG\_MAP

// Standard printing commands

ON\_COMMAND(ID\_FILE\_PRINT, CView::OnFilePrint)

ON\_COMMAND(ID\_FILE\_PRINT\_DIRECT, CView::OnFilePrint)

ON\_COMMAND(ID\_FILE\_PRINT\_PREVIEW, CView::OnFilePrintPreview)

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// CDialogsView construction/destruction

CDialogsView::CDialogsView()

{

// TODO: add construction code here

}

CDialogsView::~CDialogsView()

{

}

BOOL CDialogsView::PreCreateWindow(CREATESTRUCT& cs)

{

// TODO: Modify the Window class or styles here by modifying

// the CREATESTRUCT cs

return CView::PreCreateWindow(cs);

}

/////////////////////////////////////////////////////////////////////////////

// CDialogsView drawing

void CDialogsView::OnDraw(CDC\* pDC)

{

CDialogsDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

pDC->TextOut(0,0,pDoc->StringData);

// TODO: add draw code for native data here

}

/////////////////////////////////////////////////////////////////////////////

// CDialogsView printing

BOOL CDialogsView::OnPreparePrinting(CPrintInfo\* pInfo)

{

// default preparation

return DoPreparePrinting(pInfo);

}

void CDialogsView::OnBeginPrinting(CDC\* /\*pDC\*/, CPrintInfo\* /\*pInfo\*/)

{

// TODO: add extra initialization before printing

}

void CDialogsView::OnEndPrinting(CDC\* /\*pDC\*/, CPrintInfo\* /\*pInfo\*/)

{

// TODO: add cleanup after printing

}

/////////////////////////////////////////////////////////////////////////////

// CDialogsView diagnostics

#ifdef \_DEBUG

void CDialogsView::AssertValid() const

{

CView::AssertValid();

}

void CDialogsView::Dump(CDumpContext& dc) const

{

CView::Dump(dc);

}

CDialogsDoc\* CDialogsView::GetDocument() // non-debug version is inline

{

ASSERT(m\_pDocument->IsKindOf(RUNTIME\_CLASS(CDialogsDoc)));

return (CDialogsDoc\*)m\_pDocument;

}

#endif //\_DEBUG

/////////////////////////////////////////////////////////////////////////////

// CDialogsView message handlers

void CDialogsView::OnFileShowdialog()

{

Dlg dlg;

int result=dlg.DoModal();

if(result==IDOK)

{

CDialogsDoc\* pDoc=GetDocument();

ASSERT\_VALID(pDoc);

pDoc->StringData=dlg.m\_text;

Invalidate();

}

}

Dlg.h and Dlg.cpp

#if !defined(AFX\_DLG\_H\_\_1F53C1FE\_FF81\_4294\_8376\_0D943E38E6EA\_\_INCLUDED\_)

#define AFX\_DLG\_H\_\_1F53C1FE\_FF81\_4294\_8376\_0D943E38E6EA\_\_INCLUDED\_

#if \_MSC\_VER > 1000

#pragma once

#endif // \_MSC\_VER > 1000

// Dlg.h : header file

//

/////////////////////////////////////////////////////////////////////////////

// Dlg dialog

class Dlg : public CDialog

{

// Construction

public:

Dlg(CWnd\* pParent = NULL); // standard constructor

// Dialog Data

//{{AFX\_DATA(Dlg)

enum { IDD = IDD\_DIALOG1 };

CString m\_text;

//}}AFX\_DATA

// Overrides

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(Dlg)

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV support

//}}AFX\_VIRTUAL

// Implementation

protected:

// Generated message map functions

//{{AFX\_MSG(Dlg)

afx\_msg void OnButton1();

virtual void OnOK();

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

};

//{{AFX\_INSERT\_LOCATION}}

// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX\_DLG\_H\_\_1F53C1FE\_FF81\_4294\_8376\_0D943E38E6EA\_\_INCLUDED\_)

// Dlg.cpp : implementation file

//

#include "stdafx.h"

#include "Dialogs.h"

#include "Dlg.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

/////////////////////////////////////////////////////////////////////////////

// Dlg dialog

Dlg::Dlg(CWnd\* pParent /\*=NULL\*/)

: CDialog(Dlg::IDD, pParent)

{

//{{AFX\_DATA\_INIT(Dlg)

m\_text = \_T("");

//}}AFX\_DATA\_INIT

}

void Dlg::DoDataExchange(CDataExchange\* pDX)

{

CDialog::DoDataExchange(pDX);

//{{AFX\_DATA\_MAP(Dlg)

DDX\_Text(pDX, IDC\_EDIT1, m\_text);

//}}AFX\_DATA\_MAP

}

BEGIN\_MESSAGE\_MAP(Dlg, CDialog)

//{{AFX\_MSG\_MAP(Dlg)

ON\_BN\_CLICKED(IDC\_BUTTON1, OnButton1)

//}}AFX\_MSG\_MAP

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// Dlg message handlers

void Dlg::OnButton1()

{

m\_text=" Welcome to dialogs";

UpdateData(false);

}

void Dlg::OnOK()

{

UpdateData(true);

CDialog::OnOK();

}

**OUTPUT:**

