**Program 1:-Write a program to create a simple window**

**Source code:-**

//program to create a simple window

#include<afxwin.h>

class myframe:public CFrameWnd

{

public:

myframe()

{

Create(0,"Desired Size Window", WS\_OVERLAPPEDWINDOW,CRect(10,20,200,300));

}

};

class myapp:public CWinApp

{

public:

int InitInstance()

{

myframe \*p;

p=new myframe();

m\_pMainWnd=p;

p->ShowWindow(2);

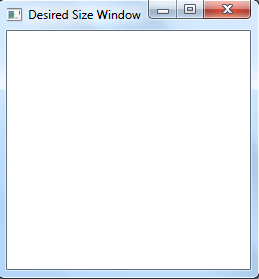
return 1;

}

};

myapp a;

**output:-**



**Program: 2: Write a program to create a window of my own class, own cursor and**

**Own icon.**

**Source Code:**

#include<afxwin.h>

#include"resource.h"

class myFrame:public CFrameWnd

{

public:

myFrame()

{

CString mywin;

HBRUSH mb;

mb=(HBRUSH)::GetStockObject(LTGRAY\_BRUSH);

mywin=AfxRegisterWndClass(CS\_HREDRAW|CS\_VREDRAW,AfxGetApp()->LoadCursor(IDC\_CURSOR1),mb,AfxGetApp()->LoadIcon(IDI\_ICON1));

Create(mywin,"Own Window type");

}

};

class mywin:public CWinApp

{

public:

int InitInstance()

{

myFrame\*p;

p=new myFrame();

m\_pMainWnd=p;

p->ShowWindow(1);

return 1;

}

};

mywin a;

**Output:**



**Program 3: Write a program to display different shapes in a window.**

**Source Code:**

#include<afxwin.h>

class myframe:public CFrameWnd

{

public:

myframe()

{

Create(0,"Standard Shapes");

}

void OnPaint()

{

CPaintDC d(this);

CBrush mybrush(RGB(150,100,0));

d.SelectObject(&mybrush);

d.MoveTo(10,10);

d.LineTo(200,10);

CBrush mybrush1(RGB(255,0,0));

d.SelectObject(&mybrush1);

d.Rectangle(10,20,200,100);

CBrush mybrush2(RGB(250,0,200));

d.SelectObject(&mybrush2);

d.RoundRect(10,120,200,220,20,20);

CBrush mybrush3(RGB(255,0,0));

d.SelectObject(&mybrush3);

d.Ellipse(10,240,200,340);

CBrush mybrush4(RGB(2,0,0));

d.SelectObject(&mybrush4);

d.Pie(250,10,350,110,350,110,350,10);

CBrush mybrush5(RGB(150,0,0));

d.SelectObject(&mybrush5);

POINT pt[5]={250,150,250,300,300,350,400,300,320,190};

d.Polygon(pt,5);

}

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(myframe,CFrameWnd)

ON\_WM\_PAINT()

END\_MESSAGE\_MAP()

class myapp:public CWinApp

{

public:

int InitInstance()

{

myframe \*p;

p=new myframe;

p->ShowWindow(3);

m\_pMainWnd=p;

return 1;

}

};

myapp a;

**Output:**



**Program 4:** **Write a Program to display origin in a middle of a window**

**Source code:**

// Program to display origin in a middle of a window

#include<afxwin.h>

class myframe : public CFrameWnd

{

public:

myframe()

{

Create(0,"standard shape");

}

void OnPaint()

{

CPaintDC d(this);

int xorigin,yorigin;

CRect r;

GetClientRect(&r);

xorigin=(r.right-r.left)/2;

yorigin=(r.bottom-r.top)/2;

d.MoveTo(0,yorigin);

d.LineTo(r.right,yorigin);

d.MoveTo(xorigin,0);

d.LineTo(xorigin,r.bottom);

d.SetViewportOrg(xorigin,yorigin);

d.SetMapMode(MM\_LOMETRIC);

d.MoveTo(-100,-100); //draw areas with origin at center

d.LineTo(100,100);

}

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(myframe,CFrameWnd)

ON\_WM\_PAINT()

END\_MESSAGE\_MAP()

class myapp:public CWinApp

{

public:

int InitInstance()

{

myframe \*p;

p=new myframe;

p->ShowWindow(3);

m\_pMainWnd=p;

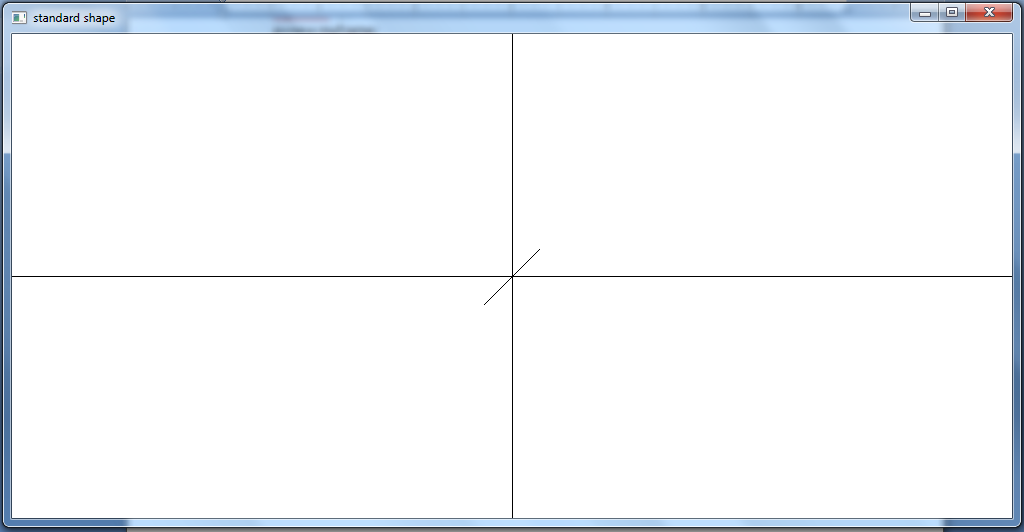
return 1;

}

};

myapp a;

**Output:**



**Program 5: Write a program to fill background of client area with bitmap**

**Source code:**

//Program of background filling with bitmap

#include<afxwin.h>

#include<afxext.h>

#include"resource.h"

class myframe:public CFrameWnd

{

public:

myframe()

{

Create(0,"Bitmap fill");

}

int OnEraseBkgnd(CDC \*d)

{

int x,y;

CRect r;

GetClientRect(&r);

CBitmap mybitmap;

CDC mymemdc;

mybitmap.LoadBitmap(IDB\_BITMAP1);

mymemdc.CreateCompatibleDC(d);

mymemdc.SelectObject(&mybitmap);

for(y=0;y<=r.bottom;y+=50)

{

for(x=0;x<=r.right;x+=30)

{

d->BitBlt(x,y,30,50,&mymemdc,0,0,SRCCOPY);

}

}

return 1;

}

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(myframe,CFrameWnd)

ON\_WM\_ERASEBKGND()

END\_MESSAGE\_MAP()

class myapp:public CWinApp

{

public:

int InitInstance()

{

myframe \*b;

b=new myframe();

m\_pMainWnd=b;

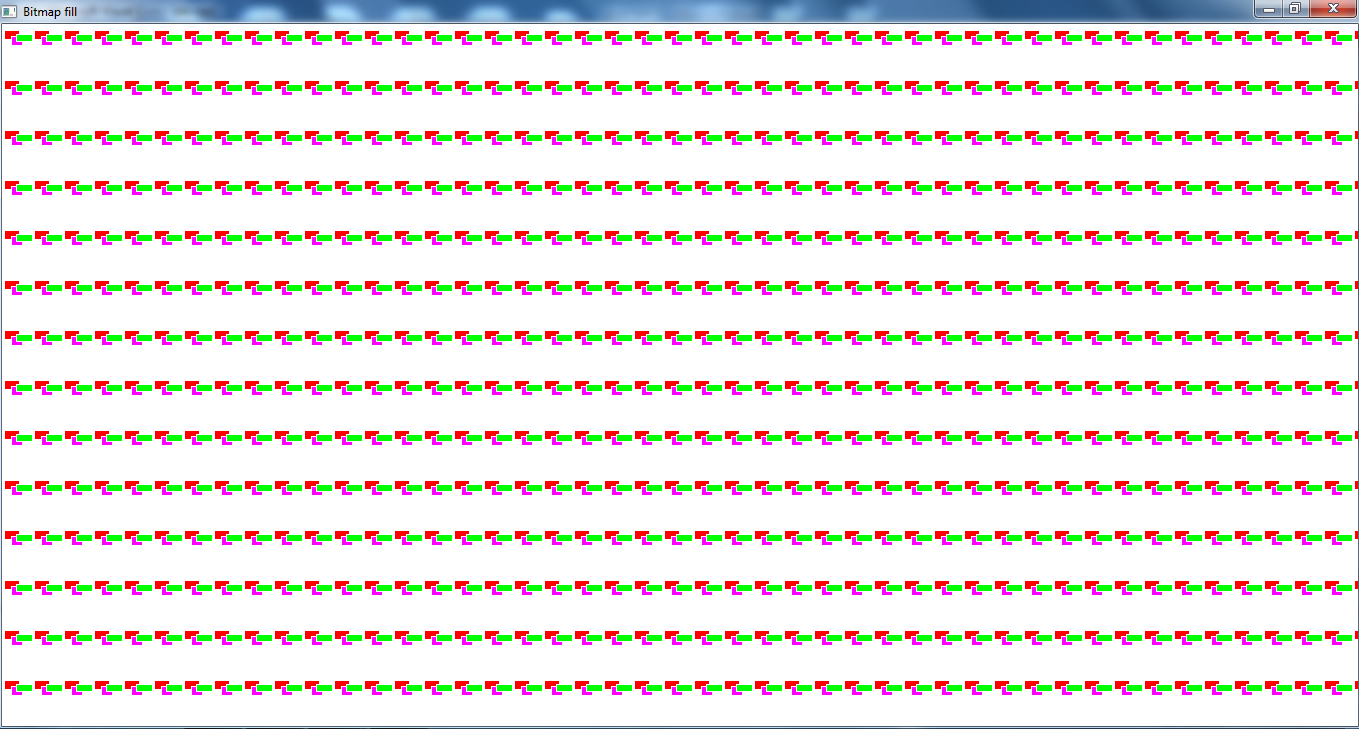
b->ShowWindow(3);

return 1;

}

};

myapp a;



**Program 6:- Write a Program to display text of my 0wn choice in a window**

**Source code:-**

//Program to display text in a window

#include<afxwin.h>

class myframe:public CFrameWnd

{

public:

myframe()

{

Create(0,"Display Text");

}

void OnPaint()

{

CPaintDC d(this);

d.TextOut(50,100,"Hello",5);

d.Rectangle(120,100,300,350);

}

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(myframe,CFrameWnd)

ON\_WM\_PAINT()

END\_MESSAGE\_MAP()

class WinApp:public CWinApp

{

int InitInstance()

{

myframe \*p;

p=new myframe();

m\_pMainWnd=p;

p->ShowWindow(3);

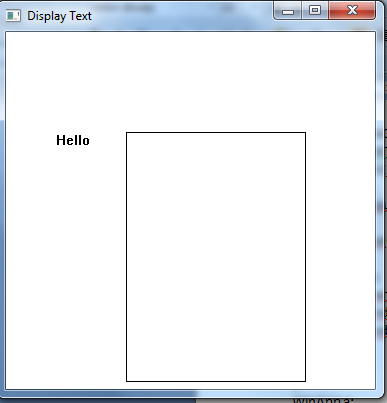
return 1;

}

};

WinApp a;

**Output:-**



**Program 7: Write a program to display menu bar,toolbar and use of keyboard accelerator in window and also interact with menu bar.**

**Source code:**

//Program of menu bar,toolbar,keyboard acccelerator

#include<afxwin.h>

#include<afxext.h>

#include "resource.h"

class myframe:public CFrameWnd

{

private:

CToolBar t;

public:

myframe()

{

LoadAccelTable(MAKEINTRESOURCE(IDR\_ACCELERATOR1));

Create(0,"Accelerator",WS\_OVERLAPPEDWINDOW,rectDefault,0,MAKEINTRESOURCE(IDR\_MENU1));

}

int OnCreate(LPCREATESTRUCT I)

{

t.Create(this, WS\_VISIBLE|WS\_CHILD|CBRS\_TOP);

t.LoadToolBar(IDR\_TOOLBAR1);

return 0;

}

void fun1()

{

MessageBox("Reached here to draw a line ","Title");

}

void fun2()

{

MessageBox("Reached here to draw a rectangle","Title");

}

void fun3()

{

MessageBox("Reached here to draw circle","Title");

}

void fun4()

{

MessageBox("Reached here to draw triangle","Title");

}

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(myframe,CFrameWnd)

ON\_WM\_CREATE()

ON\_COMMAND(101,fun1)

ON\_COMMAND(102,fun2)

ON\_COMMAND(103,fun3)

ON\_COMMAND(104,fun4)

END\_MESSAGE\_MAP()

class myapp:public CWinApp

{

public:

int InitInstance()

{

myframe\*p;

p=new myframe;

p->ShowWindow(3);

m\_pMainWnd=p;

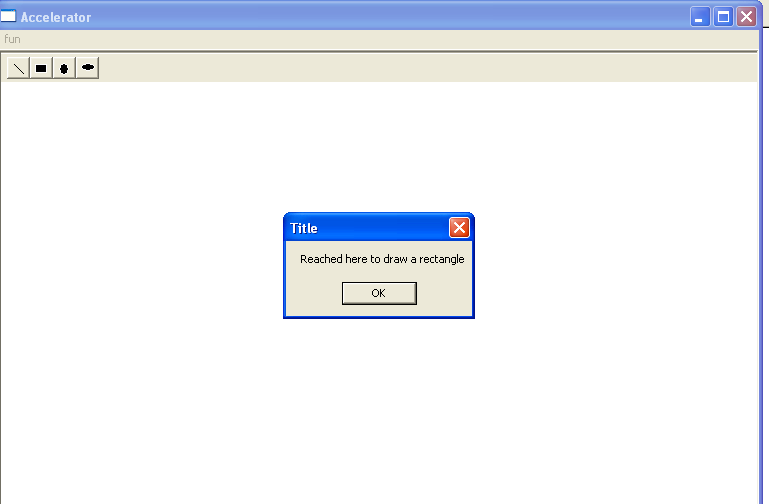
return 1;

}

};

myapp a;

**Output:**



**Program 8: Write a program to find whether a mouse is attached or not and how many button present on mouse.**

**Souce code:**

//Program whether mouse present or not

#include <afxwin.h>

class myframe:public CFrameWnd

{

public:

myframe()

{

Create(0,"Mouse Present/absent");

}

void test1()

{

int value,num;

char str[5];

value=::GetSystemMetrics(SM\_MOUSEPRESENT);

if (value==1)

{

MessageBox("Mouse is present","Relax!");

num=::GetSystemMetrics(SM\_CMOUSEBUTTONS);

sprintf(str,"%d",num);

MessageBox(str,"number of mouse buttons");

}

else

{

MessageBox("Mouse is not present","");

}

}

};

class myapp:public CWinApp

{

public:

int InitInstance()

{

myframe \*p;

p=new myframe;

p->ShowWindow(1);

p->test1();

m\_pMainWnd=p;

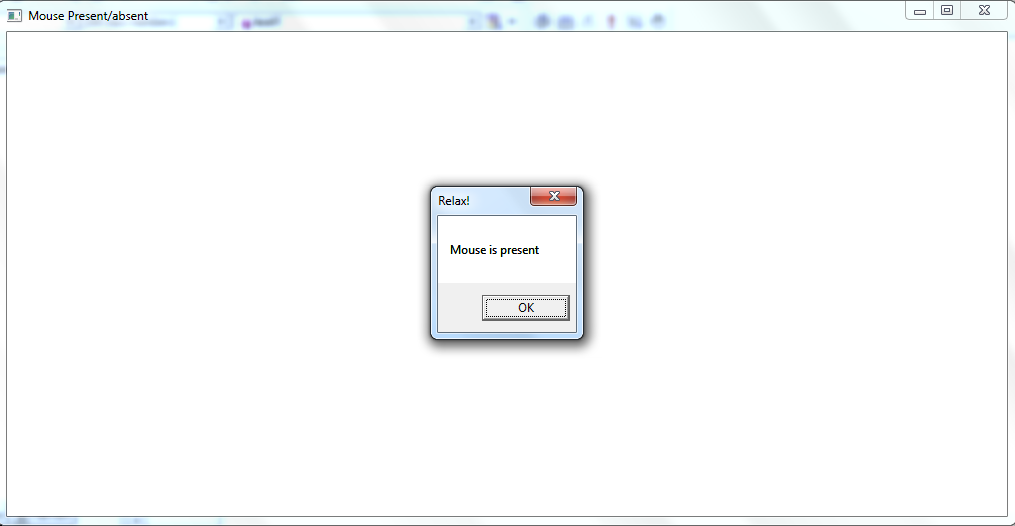
return 1;

}

};

myapp a;

**output:**



**Program: 9: Write a program to display text on left mouse button click in window.**

**Source Code:**

#include<afxwin.h>

class myframe:public CFrameWnd

{

public:

myframe()

{

Create(0,"Text Display",WS\_OVERLAPPEDWINDOW);

}

void OnLButtonDown(UINT Flag,CPoint pt)

{

CClientDC d(this);

d.TextOut(pt.x,pt.y,"puja");

}

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(myframe,CFrameWnd)

ON\_WM\_LBUTTONDOWN()

END\_MESSAGE\_MAP()

class myapp:public CWinApp

{

public:

int InitInstance()

{

myframe\*p;

p=new myframe;

p->ShowWindow(3);

m\_pMainWnd=p;

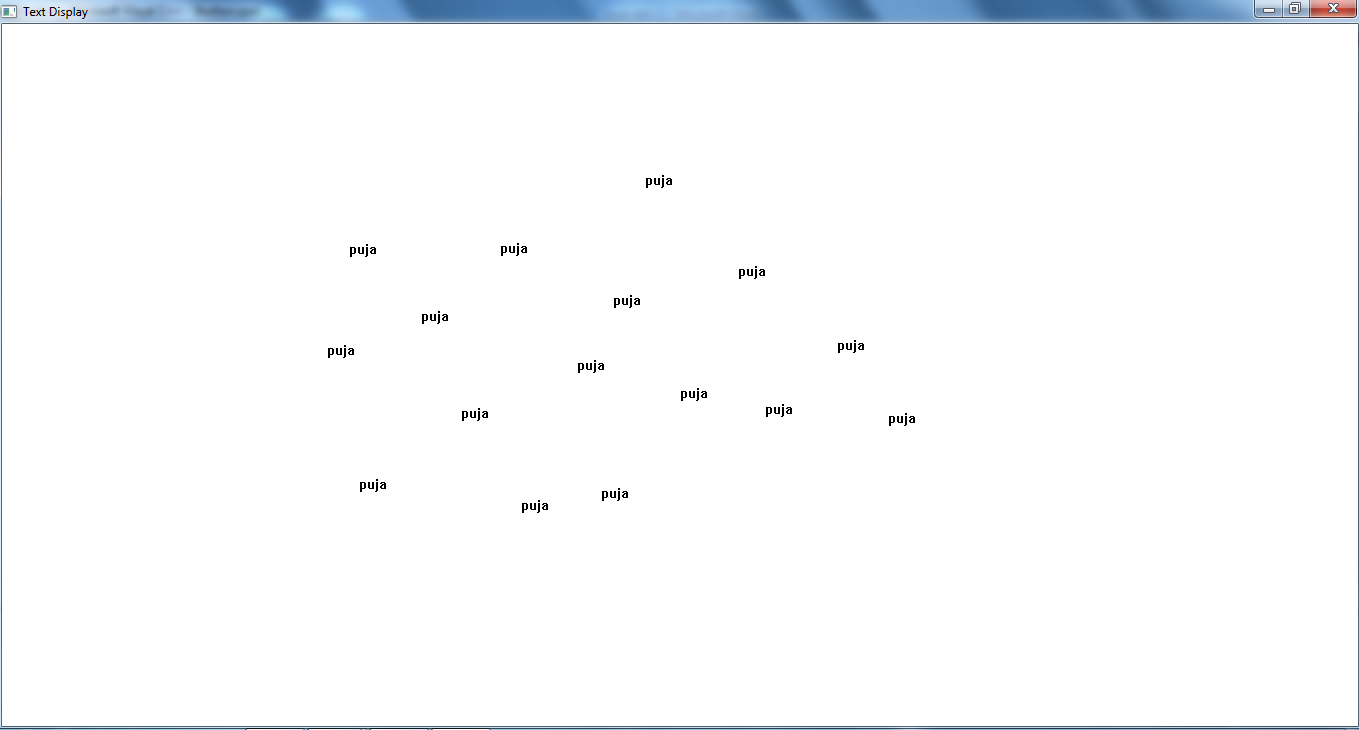
return 1;

}

};

myapp a;

**Output:**



**Program10: Write a program to display two toolbar in window.**

**Source Code:**

#include<afxwin.h>

#include<afxext.h>

#include"resource.h"

class myframe:public CFrameWnd

{

private:

CToolBar t1,t2;

CStatusBar s;

unsigned int indicators[4];

public:

myframe()

{

Create(0,"Loading Two Toolbars");

}

int OnCreate(LPCREATESTRUCT I)

{

CFrameWnd::OnCreate (I);

s.Create(this);

indicators[0]= 0;

indicators[1]=ID\_INDICATOR\_CAPS;

indicators[2]=ID\_INDICATOR\_NUM;

indicators[3]=ID\_INDICATOR\_SCRL;

s.SetIndicators(indicators,4);

t1.Create(this,WS\_CHILD|WS\_VISIBLE | CBRS\_LEFT| CBRS\_BORDER\_LEFT|CBRS\_BORDER\_RIGHT|CBRS\_BORDER\_TOP|CBRS\_BORDER\_BOTTOM);

t1.LoadToolBar(IDR\_TOOLBAR1);

t2.Create(this,WS\_CHILD | WS\_VISIBLE | CBRS\_RIGHT| CBRS\_BORDER\_LEFT | CBRS\_BORDER\_RIGHT | CBRS\_BORDER\_TOP|CBRS\_BORDER\_BOTTOM);

t2.LoadToolBar(IDR\_TOOLBAR2);

return 0;

}

void drawline()

{

MessageBox("Line","drawline");

}

void drawrectangle()

{

MessageBox("Rectangle","drawrectangle");

}

void drawcircle()

{

MessageBox("Circle","drawcircle");

}

void drawtriangle()

{

MessageBox("Triangle","drawtriangle");

}

void redcolor()

{

MessageBox("Red","redcolor");

}

void greencolor()

{

MessageBox("Green","greencolor");

}

void bluecolor()

{

MessageBox("Blue","bluecolor");

}

void yellowcolor()

{

MessageBox("Yellow","yellowcolor");

}

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(myframe,CFrameWnd)

ON\_WM\_CREATE()

ON\_COMMAND(101,drawline)

ON\_COMMAND(102,drawrectangle)

ON\_COMMAND(103,drawcircle)

ON\_COMMAND(104,drawtriangle)

ON\_COMMAND(201,redcolor)

ON\_COMMAND(202,greencolor)

ON\_COMMAND(203,bluecolor)

ON\_COMMAND(204,yellowcolor)

END\_MESSAGE\_MAP()

class myapp:public CWinApp

{

public:

int InitInstance()

{

myframe \*p;

p=new myframe;

p->ShowWindow(3);

m\_pMainWnd=p;

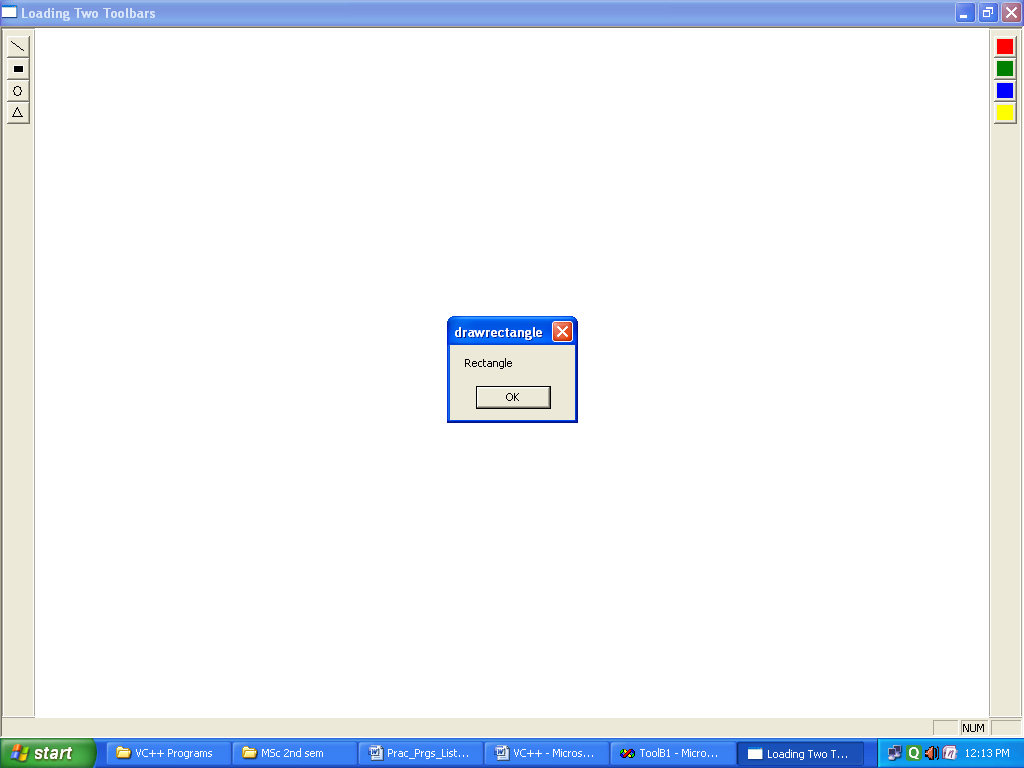
return 1;

}

};

myapp a;

**Output:**

****

**Program11: Write a program to display status bar in a window and display status of**

**Toggle keys.**

**Source Code:**

#include<afxwin.h>

#include<afxext.h>

class myframe:public CFrameWnd

{

private:

CStatusBar S;

unsigned int i[4];

public:

myframe()

{

Create (0,"status Bar");

}

int OnCreate(LPCREATESTRUCT I)

{

S.Create(this);

i[0]=0;

i[1]=ID\_INDICATOR\_CAPS;

i[2]=ID\_INDICATOR\_NUM;

i[3]=ID\_INDICATOR\_SCRL;

S.SetIndicators(i,4);

return 0;

}

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(myframe,CFrameWnd)

ON\_WM\_CREATE()

END\_MESSAGE\_MAP()

class myapp:public CWinApp

{

public:

int InitInstance()

{

myframe\*p;

p=new myframe();

p->ShowWindow(3);

m\_pMainWnd=p;

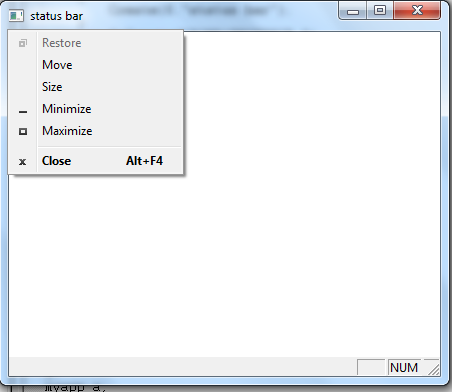
return 1;

}

};

myapp a;

**Output:**

****

**Program12: Write a Program to create simple dialog box and interact with it**

**Source code:**

//program to create simple dialog box

#include<afxwin.h>

#include<afxdlgs.h>

#include"resource.h"

class mydialog:public CDialog

{

public:

mydialog(int n):CDialog(n)

{

}

void OnOK()

{

CDialog:: OnOK();

MessageBox("Ok Button","Ok Handler");

}

void OnCancel()

{

CDialog::OnCancel();

MessageBox("Cancel Button","Cancle Handler");

}

};

class myframe:public CFrameWnd

{

public:

myframe()

{

Create(0,"Dialog display",WS\_OVERLAPPEDWINDOW,rectDefault,0,MAKEINTRESOURCE(IDR\_MENU1));

}

void about()

{

mydialog d(IDD\_DIALOG1);

d.DoModal();

}

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(myframe,CFrameWnd)

ON\_COMMAND(101,about)

END\_MESSAGE\_MAP()

class myapp: public CWinApp

{

public:

int InitInstance()

{

myframe \*p;

p=new myframe();

p->ShowWindow(3);

m\_pMainWnd=p;

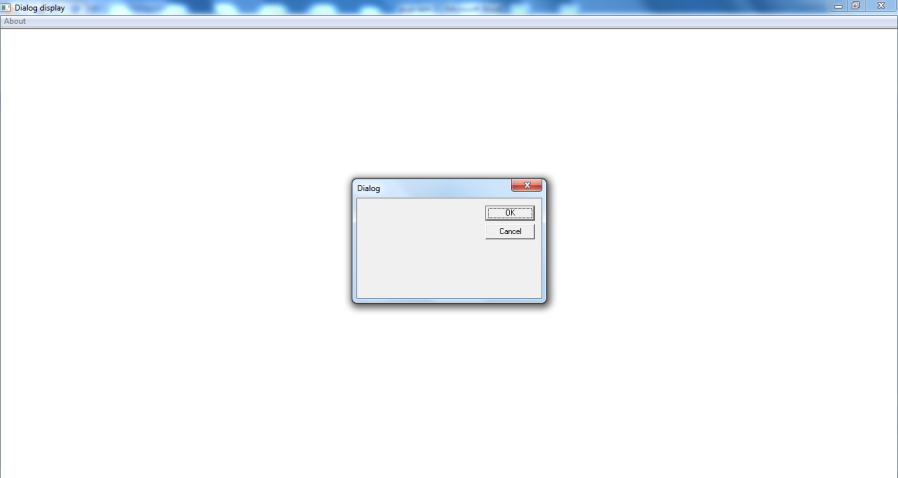
return 1;

}

};

myapp a;

**Output:**



**Program 13: Write a Program to validate the data entered in a dialog box**

**Source code:**

//Program to validate the data entered in a dialog box

#include<afxwin.h>

#include<afxdlgs.h>

#include"resource.h"

class mydialog:public CDialog

{

private:

CString bn;

float pr;

unsigned int pages;

public:

mydialog(int n):CDialog(n)

{

}

int OnInitDialog()

{

bn="Enter book name";

pr=100;

pages=100;

return CDialog::OnInitDialog();

}

void DoDataExchange(CDataExchange \*pdx)

{

CDialog :: DoDataExchange(pdx);

DDX\_Text(pdx,IDC\_NAME,bn);

DDV\_MaxChars(pdx,bn,25);

DDX\_Text(pdx,IDC\_PRICE,pr);

DDV\_MinMaxFloat(pdx,pr,50.0f,999.9f);

DDX\_Text(pdx,IDC\_PAGES,pages);

DDV\_MinMaxInt(pdx,pages,50,9999);

}

void OnOK()

{

if(UpdateData (TRUE))

{

CDialog::EndDialog(IDOK);

char s[20];

CString str;

str="Name:";

str+=bn;

str+="\n";

//str=s;

str+="Price:";

sprintf(s,"%f",pr);

str+=s;

str+="\n";

str+="Pages:";

sprintf(s,"%u",pages);

str+=s;

MessageBox(str,"DDV");

}

}

void OnCancle()

{

CDialog::OnCancel();

}

};

class myframe:public CFrameWnd

{

public:

myframe()

{

Create(0,"Dialog display",WS\_OVERLAPPEDWINDOW,rectDefault,0,MAKEINTRESOURCE(IDR\_MENU1));

}

void about()

{

mydialog d(IDD\_DIALOG1);

d.DoModal();

}

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(myframe,CFrameWnd)

ON\_COMMAND(101,about)

END\_MESSAGE\_MAP()

class myapp:public CWinApp

{

public:

int InitInstance()

{

myframe \*a;

a=new myframe();

a->ShowWindow(3);

m\_pMainWnd=a;

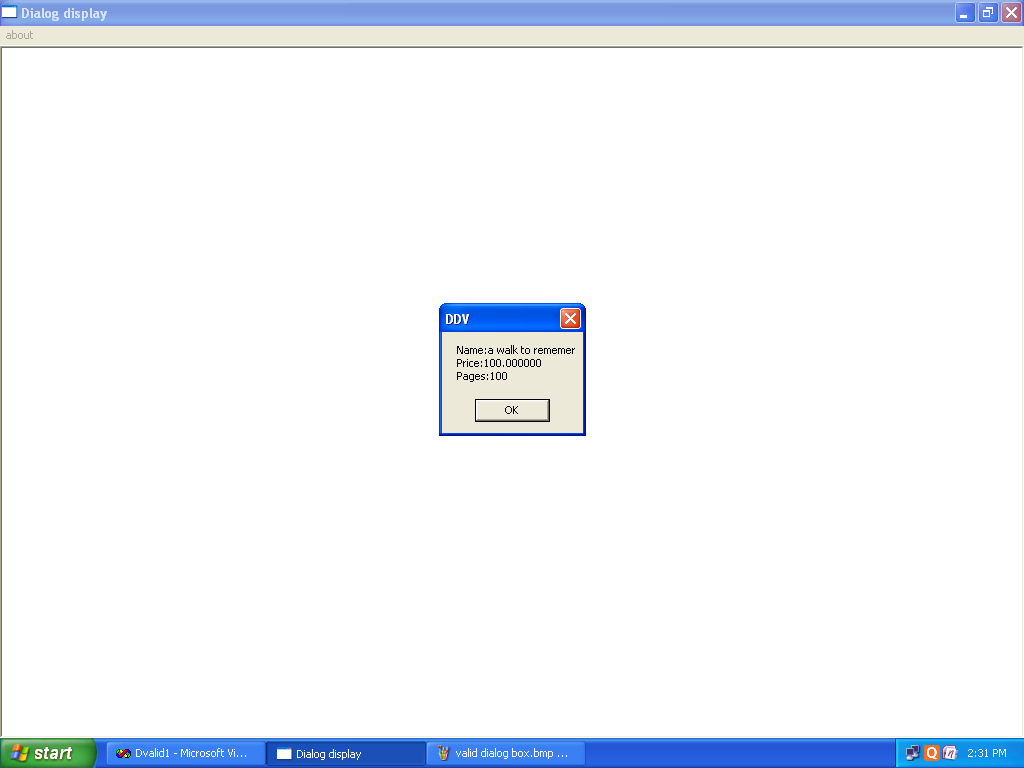
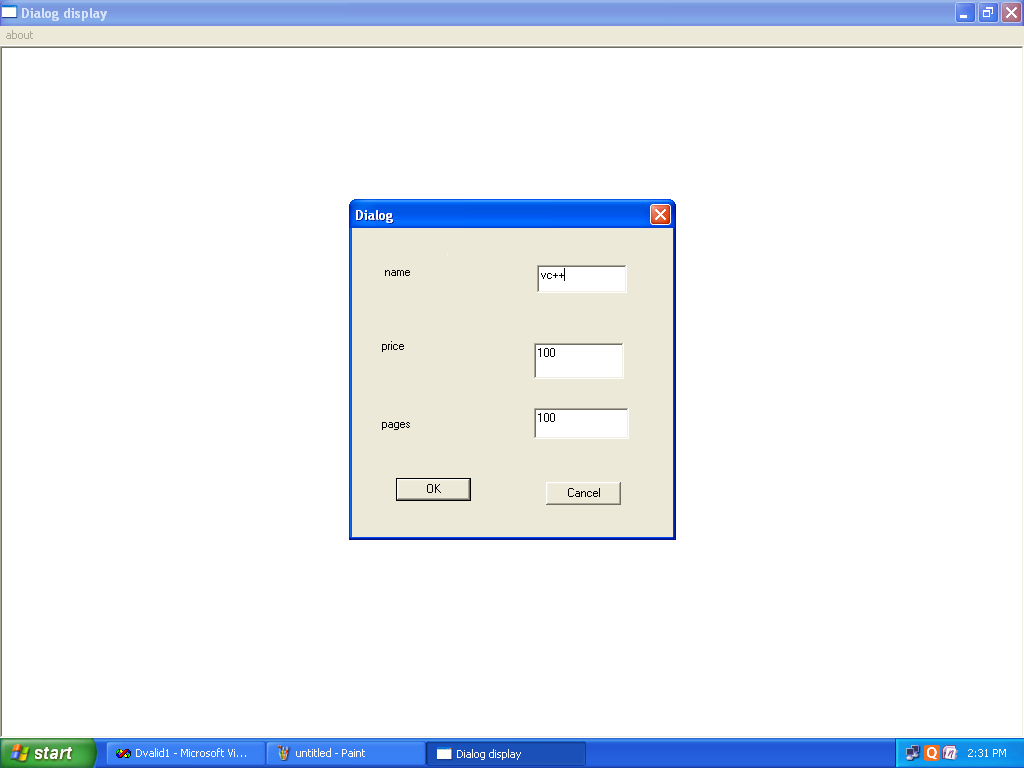
return 1;

}

};

myapp p;

**Output:**



**Program 14: Write a program to invoke common color dialog box select color from it.**

**Source code:**

//Program to display color dialog

#include<afxwin.h>

#include<afxdlgs.h>

#include "resource.h"

class myframe:public CFrameWnd

{

private:

COLORREF clr;

public:

myframe()

{

Create(0,"Color Dialog",WS\_OVERLAPPEDWINDOW,rectDefault,0,MAKEINTRESOURCE(IDR\_MENU1));

clr= RGB(255,255,255);

}

void OnPaint()

{

CPaintDC pdc (this);

CRect r;

CBrush mybrush(clr);

GetClientRect(&r);

pdc.FillRect(&r,&mybrush);

}

void color()

{

CColorDialog cd;

CRect r;

CClientDC pdc(this);

if(cd.DoModal()==IDOK)

{

clr=cd.GetColor();

CBrush mybrush(clr);

GetClientRect(&r);

pdc.FillRect(&r,&mybrush);

}

}

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(myframe, CFrameWnd)

ON\_COMMAND(101,color)

ON\_WM\_PAINT()

END\_MESSAGE\_MAP()

class myapp:public CWinApp

{

public:

int InitInstance()

{

myframe \*p;

p=new myframe;

p->ShowWindow(3);

m\_pMainWnd=p;

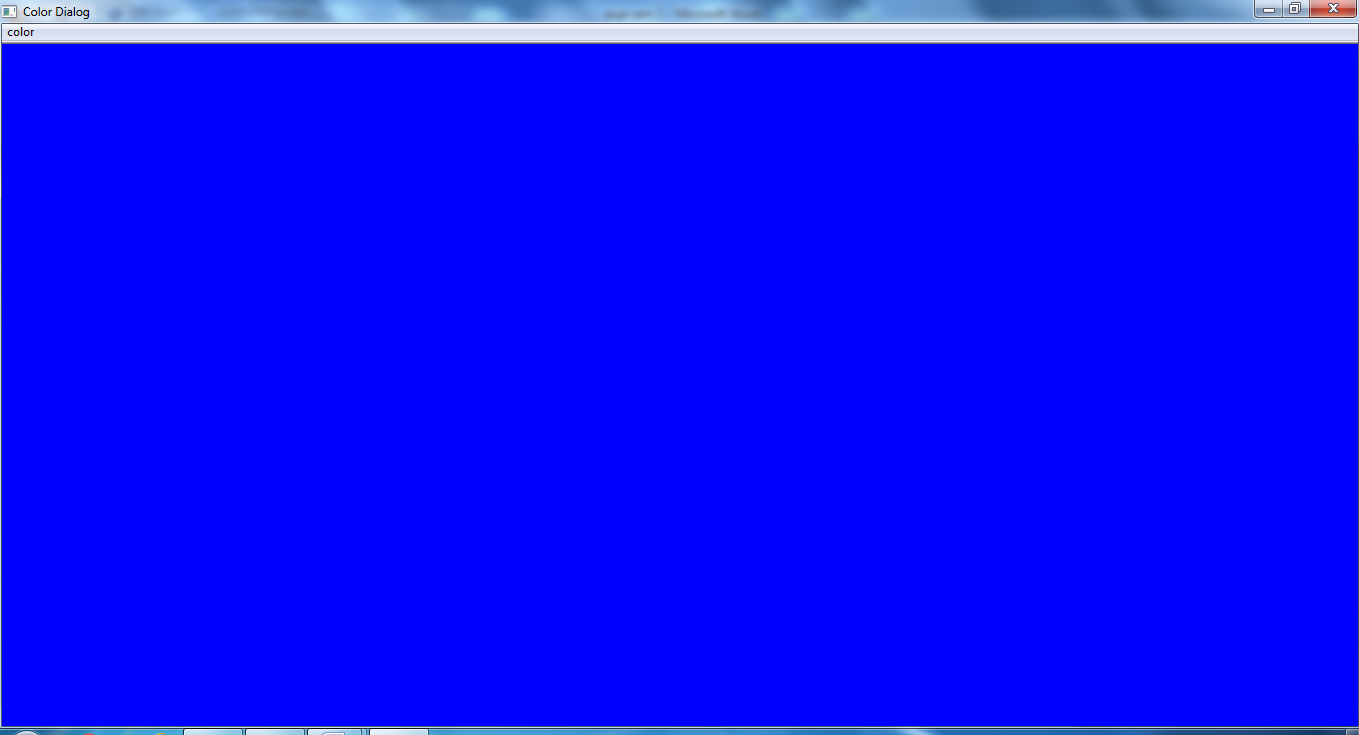
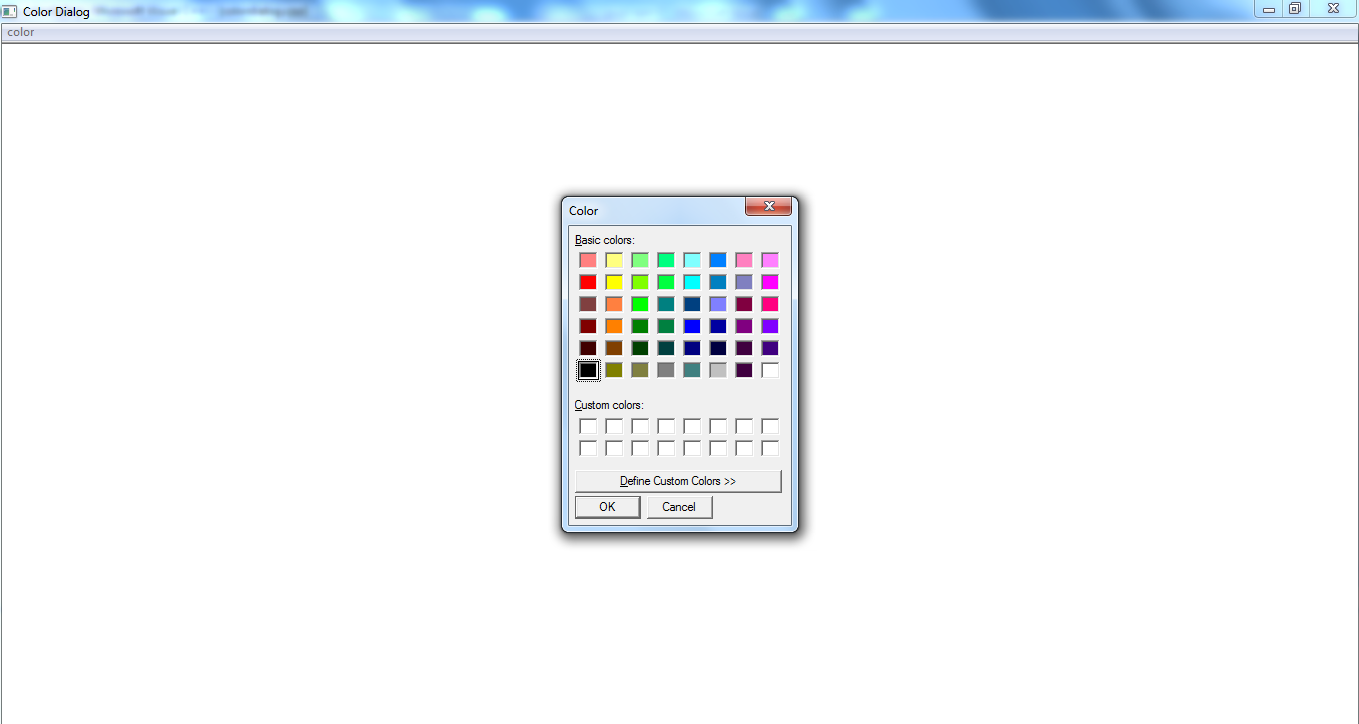
return 1;

}

};

myapp a;

**Output:**



**Program 15: Write a program to display your name in MDI application using document and view architecture.**

**Source code:-**

void CMDINAME1View::OnPaint()

{

CPaintDC dc(this); // device context for painting

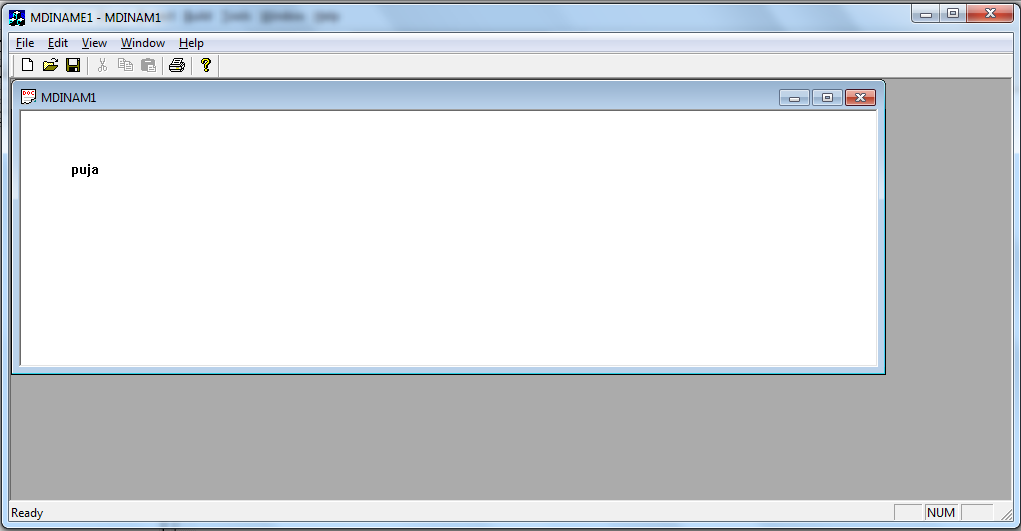
dc.TextOut(50,50,"puja",4);

// TODO: Add your message handler code here

// Do not call CView::OnPaint() for painting messages

}

**Output:**



**Program 16: Write a program to display rectangle in SDI application without using document and view architecture.**

**Souce code:**

void CChildView::OnPaint()

{

CPaintDC dc(this); // device context for painting

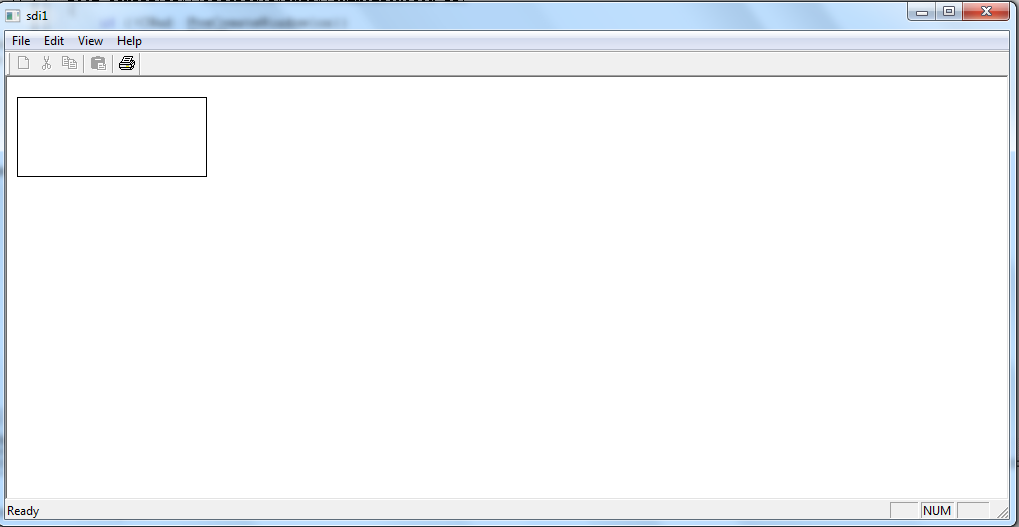
dc.Rectangle(10,20,200,100);

// TODO: Add your message handler code here

// Do not call CWnd::OnPaint() for painting messages

}

**Output:**

****

**Program 17: Write a program to display notepad using Appwizard.**

**Source code:**

// Notpad 1View.h : interface of the CNotpad1View class

//

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

#if !defined(AFX\_NOTPAD1VIEW\_H\_\_FE8D0CB8\_ABC1\_4514\_9370\_0962BEC94742\_\_INCLUDED\_)

#define AFX\_NOTPAD1VIEW\_H\_\_FE8D0CB8\_ABC1\_4514\_9370\_0962BEC94742\_\_INCLUDED\_

#if \_MSC\_VER > 1000

#pragma once

#endif // \_MSC\_VER > 1000

class CNotpad1View : public CEditView

{

protected: // create from serialization only

CNotpad1View();

DECLARE\_DYNCREATE(CNotpad1View)

// Attributes

public:

CNotpad1Doc\* GetDocument();

// Operations

public:

// Overrides

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(CNotpad1View)

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 ~CNotpad1View();

#ifdef \_DEBUG

virtual void AssertValid() const;

virtual void Dump(CDumpContext& dc) const;

#endif

protected:

// Generated message map functions

protected:

//{{AFX\_MSG(CNotpad1View)

// 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()

};

#ifndef \_DEBUG // debug version in Notpad 1View.cpp

inline CNotpad1Doc\* CNotpad1View::GetDocument()

{ return (CNotpad1Doc\*)m\_pDocument; }

#endif

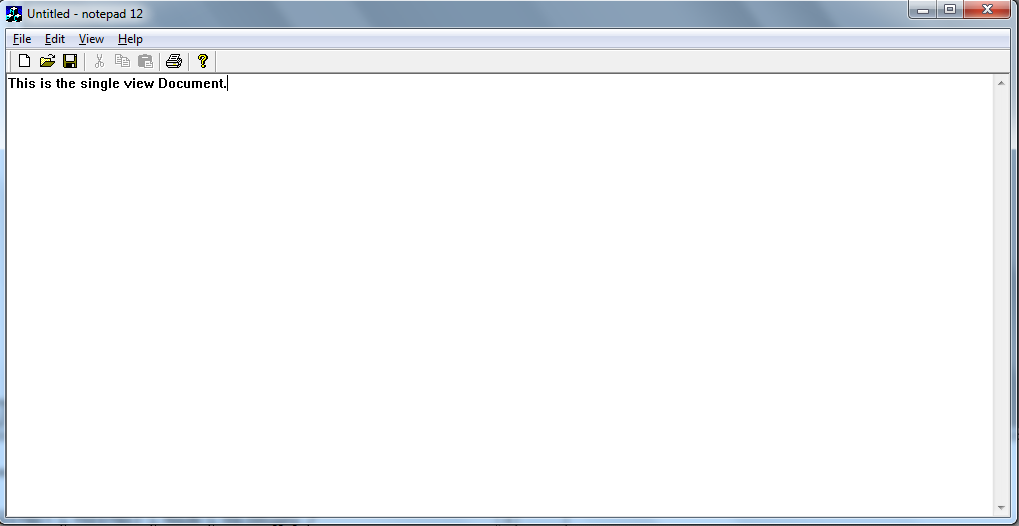
/////////////////////////////////////////////////////////////////////////////

//{{AFX\_INSERT\_LOCATION}}

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

#endif // !defined(AFX\_NOTPAD1VIEW\_H\_\_FE8D0CB8\_ABC1\_4514\_9370\_0962BEC94742\_\_INCLUDED\_)

Output:-



**Program18: Write a program to display splitter window using AppWizard.**

**Source code:-**

// CSplit1View message handlers

void CSplit1View::OnPaint()

{

CPaintDC dc(this); // device context for painting

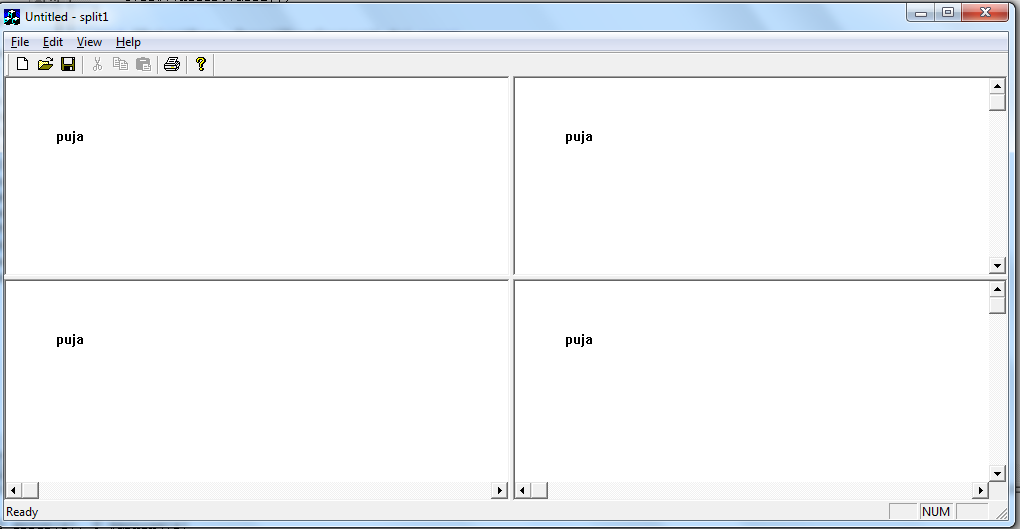
dc.TextOut(50,50,"puja",4);

// TODO: Add your message handler code here

// Do not call CView::OnPaint() for painting messages

}

**Output:-**



**Program 19: Write a program to create your own DLL make use of it in your own application.**

**Source code:-**

**(.exe appwizaed code)**

**Header file (factdemodlg.h):**

class CFactdemoDlg : public CDialog

{

// Construction

public:

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

// Dialog Data

//{{AFX\_DATA(CFactdemoDlg)

enum { IDD = IDD\_FACTDEMO\_DIALOG };

int m\_input;

double m\_output;

//}}AFX\_DATA

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(CFactdemoDlg)

protected:

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

//}}AFX\_VIRTUAL

// Implementation

protected:

HICON m\_hIcon;

// Generated message map functions

//{{AFX\_MSG(CFactdemoDlg)

virtual BOOL OnInitDialog();

afx\_msg void OnSysCommand(UINT nID, LPARAM lParam);

afx\_msg void OnPaint();

afx\_msg HCURSOR OnQueryDragIcon();

afx\_msg void OnFact();

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

};

//{{AFX\_INSERT\_LOCATION}}

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

#endif // !defined(AFX\_FACTDEMODLG\_H\_\_49004451\_10B4\_48EC\_9EC5\_DE193A9C72F8\_\_INCLUDED\_)

extern "C" \_\_declspec(dllimport) double fact11(int d);

**Command button (factdemodlg.cpp):**

void CFactdemoDlg::OnFact()

{

// TODO: Add your control notification handler code here

UpdateData(TRUE);

m\_op=fact11(m\_ip);

UpdateData(FALSE);

}

(Fact11.cpp) :-

CFact11App::CFact11App()

{

// TODO: add construction code here,

// Place all significant initialization in InitInstance

}

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

**(.dll appwizard code)**

// The one and only CFact11App object

CFact11App theApp;

extern "C" \_\_declspec(dllexport) double fact11(int a)

{

int i;

double fact;

fact=1;

for(i=1;i<=a;i++)

{

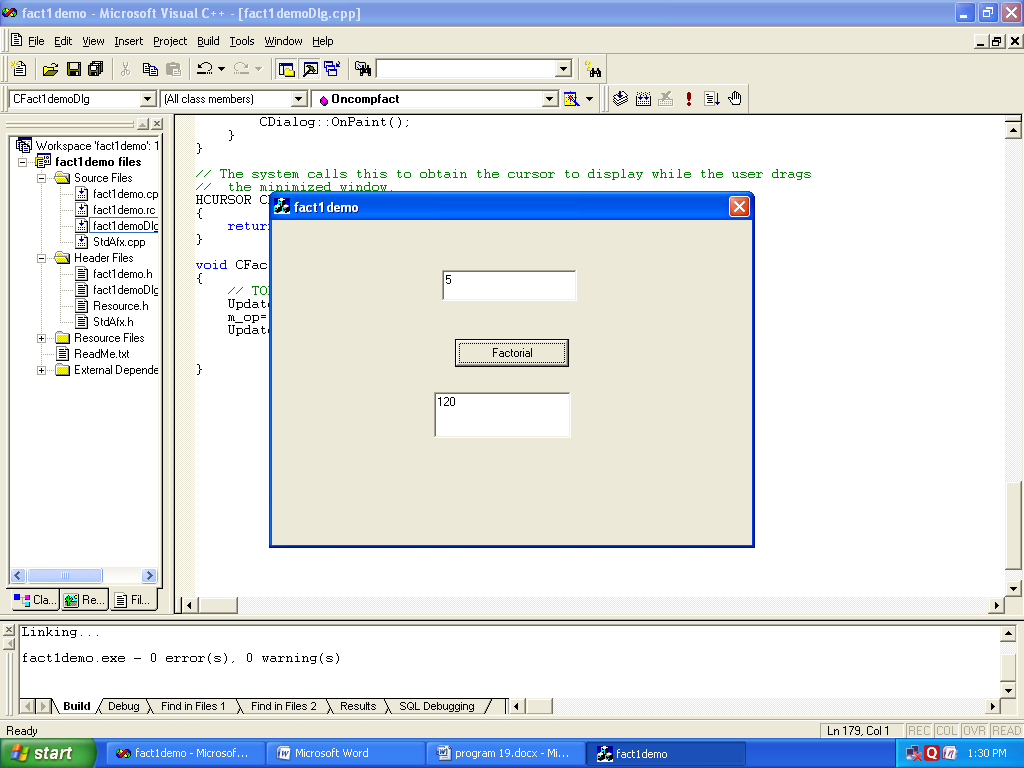
fact=fact\*i;

}

return fact;

}

**0utput:**



**Program 20**

**Program name: write a program to create library management application using odbc connection.**

**Source code:**

// mydatabaseView.cpp : implementation of the CMydatabaseView class

#include "stdafx.h"

#include "mydatabase.h"

#include "mydatabaseSet.h"

#include "mydatabaseDoc.h"

#include "mydatabaseView.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

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

#endif

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

// CMydatabaseView

IMPLEMENT\_DYNCREATE(CMydatabaseView, CRecordView)

BEGIN\_MESSAGE\_MAP(CMydatabaseView, CRecordView)

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

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

ON\_BN\_CLICKED(IDC\_BUTTON2, OnUPDATE)

ON\_BN\_CLICKED(IDC\_BUTTON3, OnDELETE)

ON\_BN\_CLICKED(IDC\_BUTTON4, OnCLEAR)

//}}AFX\_MSG\_MAP

// Standard printing commands

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

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

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

END\_MESSAGE\_MAP()

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

// CMydatabaseView construction/destruction

CMydatabaseView::CMydatabaseView()

: CRecordView(CMydatabaseView::IDD)

{

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

m\_pSet = NULL;

//}}AFX\_DATA\_INIT

// TODO: add construction code here

}

CMydatabaseView::~CMydatabaseView()

{

}

void CMydatabaseView::DoDataExchange(CDataExchange\* pDX)

{

CRecordView::DoDataExchange(pDX);

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

DDX\_FieldText(pDX, IDC\_EDIT1, m\_pSet->m\_student\_id, m\_pSet);

DDX\_FieldText(pDX, IDC\_EDIT2, m\_pSet->m\_student\_nm, m\_pSet);

DDX\_FieldText(pDX, IDC\_EDIT3, m\_pSet->m\_Book\_nm, m\_pSet);

DDX\_FieldText(pDX, IDC\_EDIT4, m\_pSet->m\_Book\_Author, m\_pSet);

DDX\_FieldText(pDX, IDC\_EDIT5, m\_pSet->m\_Book\_prize, m\_pSet);

DDX\_FieldText(pDX, IDC\_EDIT6, m\_pSet->m\_Dt\_issue, m\_pSet);

DDX\_FieldText(pDX, IDC\_EDIT7, m\_pSet->m\_Dt\_return, m\_pSet);

//}}AFX\_DATA\_MAP

}

BOOL CMydatabaseView::PreCreateWindow(CREATESTRUCT& cs)

{

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

// the CREATESTRUCT cs

return CRecordView::PreCreateWindow(cs);

}

void CMydatabaseView::OnInitialUpdate()

{

m\_pSet = &GetDocument()->m\_mydatabaseSet;

CRecordView::OnInitialUpdate();

GetParentFrame()->RecalcLayout();

ResizeParentToFit();

}

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

// CMydatabaseView printing

BOOL CMydatabaseView::OnPreparePrinting(CPrintInfo\* pInfo)

{

// default preparation

return DoPreparePrinting(pInfo);

}

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

{

// TODO: add extra initialization before printing

}

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

{

// TODO: add cleanup after printing

}

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

// CMydatabaseView diagnostics

#ifdef \_DEBUG

void CMydatabaseView::AssertValid() const

{

CRecordView::AssertValid();

}

void CMydatabaseView::Dump(CDumpContext& dc) const

{

CRecordView::Dump(dc);

}

CMydatabaseDoc\* CMydatabaseView::GetDocument() // non-debug version is inline

{

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

return (CMydatabaseDoc\*)m\_pDocument;

}

#endif //\_DEBUG

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

// CMydatabaseView database support

CRecordset\* CMydatabaseView::OnGetRecordset()

{

return m\_pSet;

}

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

// CMydatabaseView message handlers

void CMydatabaseView::OnADD()

{

// TODO: Add your control notification handler code here

m\_pSet->AddNew();

UpdateData(TRUE);

if(m\_pSet->CanUpdate())

{

m\_pSet->Update();

}

if(!m\_pSet->IsEOF())

{

m\_pSet->MoveLast();

}

m\_pSet->Requery();

UpdateData(FALSE);

}

void CMydatabaseView::OnUPDATE()

{

// TODO: Add your control notification handler code here

m\_pSet->Edit();

UpdateData(TRUE);

if(m\_pSet->CanUpdate())

{

m\_pSet->Update();

}

}

void CMydatabaseView::OnDELETE()

{

// TODO: Add your control notification handler code here

CRecordsetStatus status;

try{

m\_pSet->Delete();

}

catch(CDBException\* e)

{

AfxMessageBox(e->m\_strError);

e->Delete();

m\_pSet->MoveFirst();

UpdateData(FALSE);

return;

}

m\_pSet->GetStatus(status);

if(status.m\_lCurrentRecord==0)

{

m\_pSet->MoveFirst();

}

else

{

m\_pSet->MoveNext();

}

UpdateData(FALSE);

}

void CMydatabaseView::OnCLEAR()

{

// TODO: Add your control notification handler code here

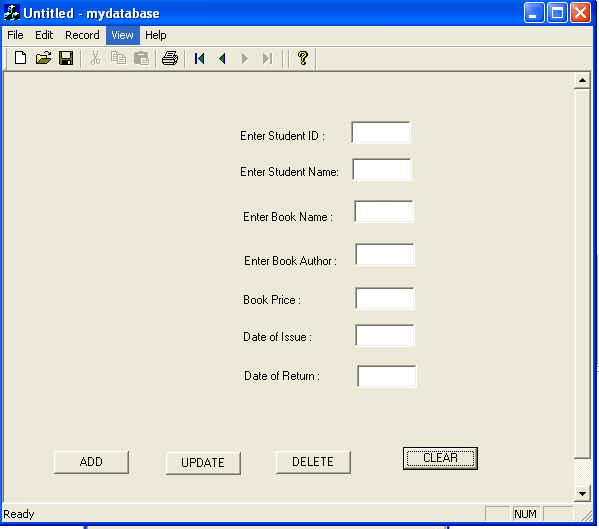
m\_pSet->SetFieldNull(NULL);

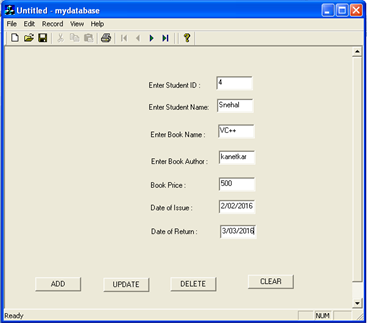
UpdateData(FALSE);

}

**Output:-**

| **mydb** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **student\_id** | **student\_nm** | **Book\_nm** | **Book\_Author** | **Book\_prize** | **Dt\_issue** | **Dt\_return** |
| 1 | megha | DMS | kolman | 450 | 02/11/2015 | 02/01/2016 |
| 2 | manish | CG | Kanetkar | 600 | 03/03/2016 | 21/03/2016 |
| 3 | sachin | VC++ | Heyes | 550 | 02/06/2016 | 10/06/2016 |





| **mydb** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **student\_id** | **student\_nm** | **Book\_nm** | **Book\_Author** | **Book\_prize** | **Dt\_issue** | **Dt\_return** |
| 1 | megha | DMS | kolman | 450 | 02/11/2015 | 02/01/2016 |
| 2 | manish | CG | Kanetkar | 600 | 03/03/2016 | 21/03/2016 |
| 3 | sachin | VC++ | Heyes | 550 | 02/06/2016 | 10/06/2016 |
| 4 | Snehal | VC++ | kanetkar | 500 | 02/02/2016 | 03/03/2016 |

**21 . write a program to create Big Bazar management application using ODBC connection.**

**Add**

void CDIV4View::OnAdd()

{

m\_pSet->AddNew();

UpdateData(TRUE);

if(m\_pSet->CanUpdate())

{

m\_pSet->Update();

}

if(!m\_pSet->IsEOF())

{

m\_pSet->MoveLast();

}

UpdateData(FALSE);

}

// TODO: Add your control notification handler code here

**//Delete**

void CDIV4View::OnDelete()

{

CRecordsetStatus status;

try

{

m\_pSet->Delete();

}

catch(CDBException \*e)

{

AfxMessageBox(e->m\_strError);

e->Delete();

m\_pSet->MoveFirst();

UpdateData();

(FALSE);

return;

}

m\_pSet->GetStatus(status);

if(status.m\_lCurrentRecord==0)

{

m\_pSet->MoveFirst();}

else

{

m\_pSet->MoveNext();

}

UpdateData(FALSE);

// TODO: Add your control notification handler code here

}

**Clear**

void CDIV4View::OnClear()

{

m\_pSet->SetFieldNull(NULL);

UpdateData(FALSE);

}

**Update**

void CDIV4View::Onupdate()

{

m\_pSet->Edit();

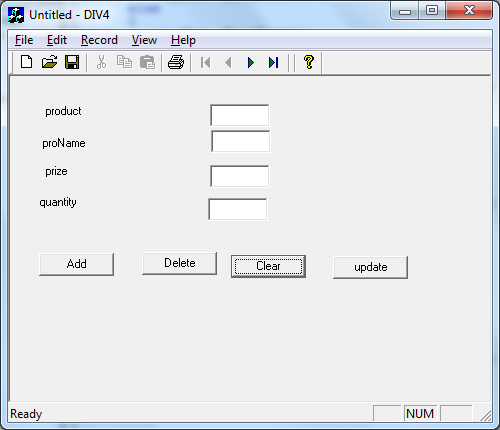
UpdateData(TRUE);

if(m\_pSet->CanUpdate())

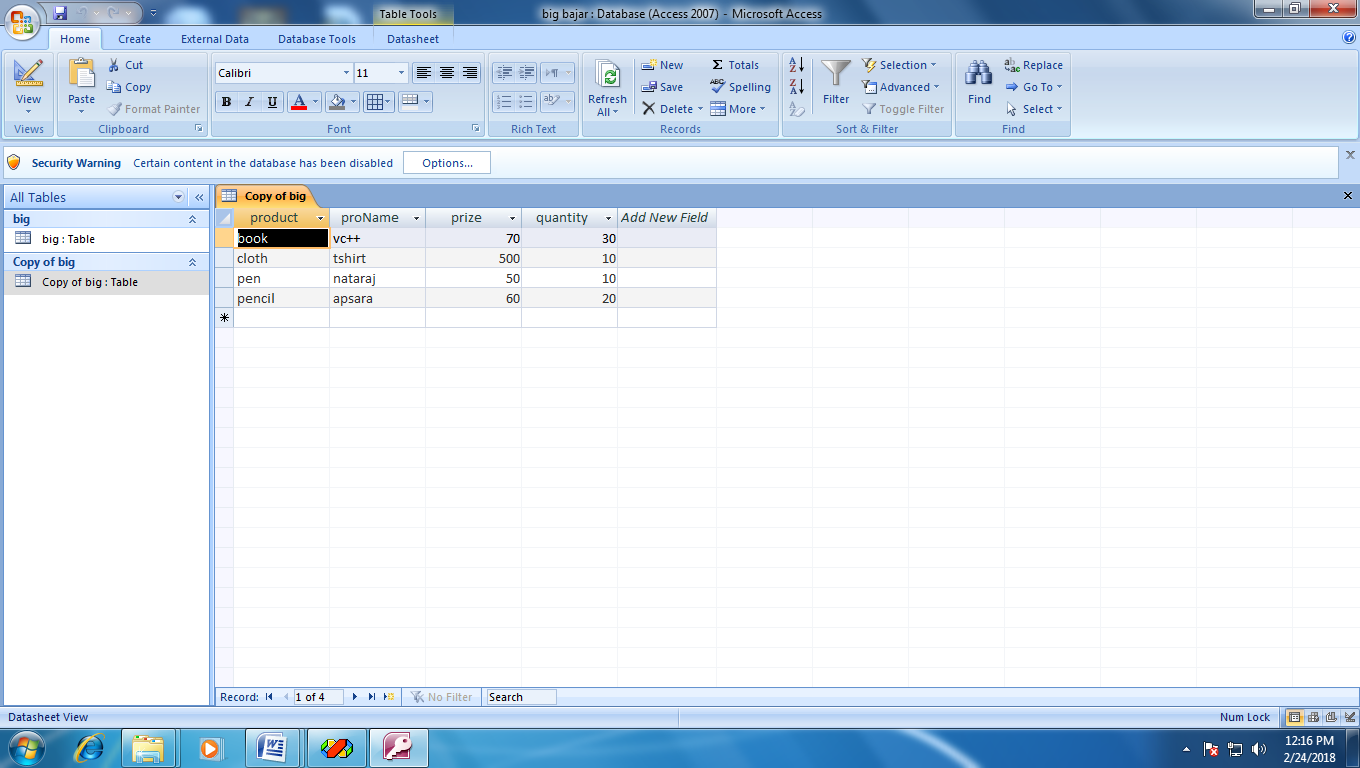
{

m\_pSet->Update();

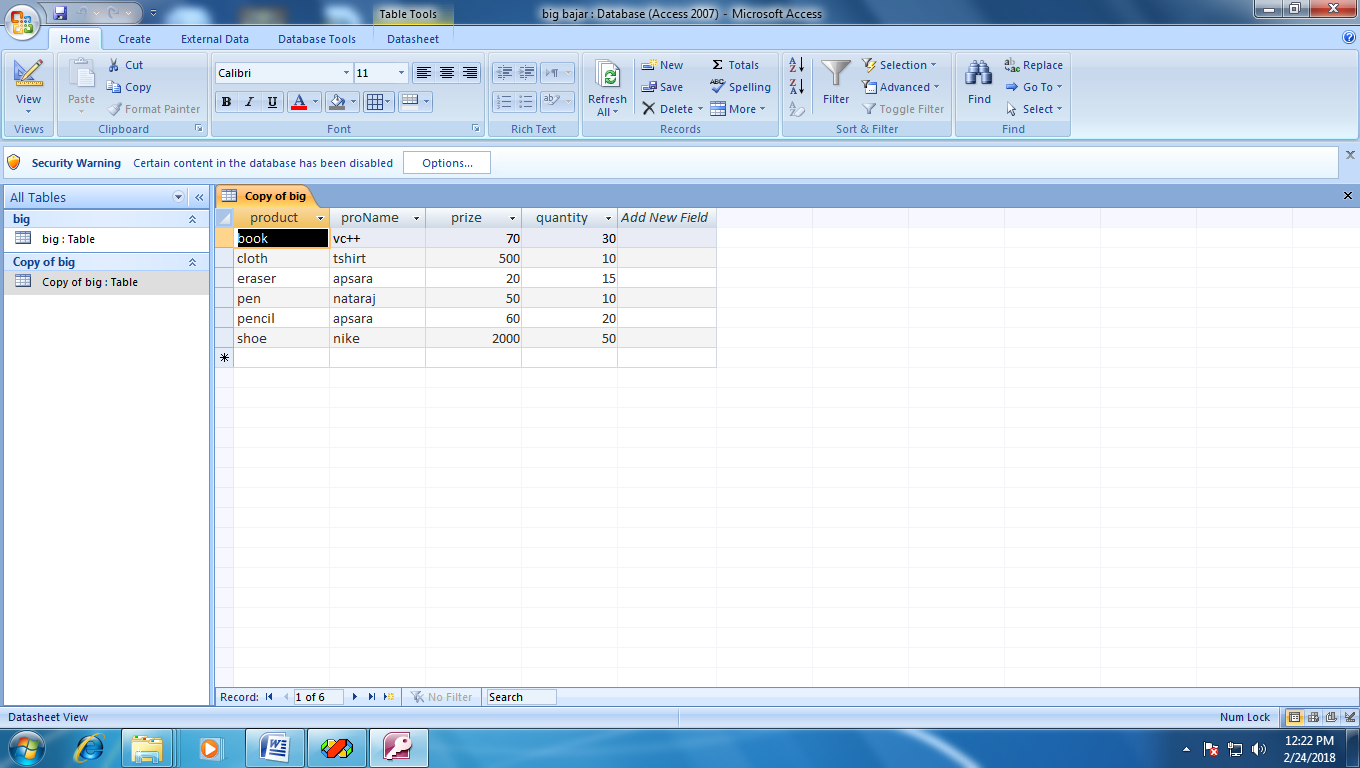
}



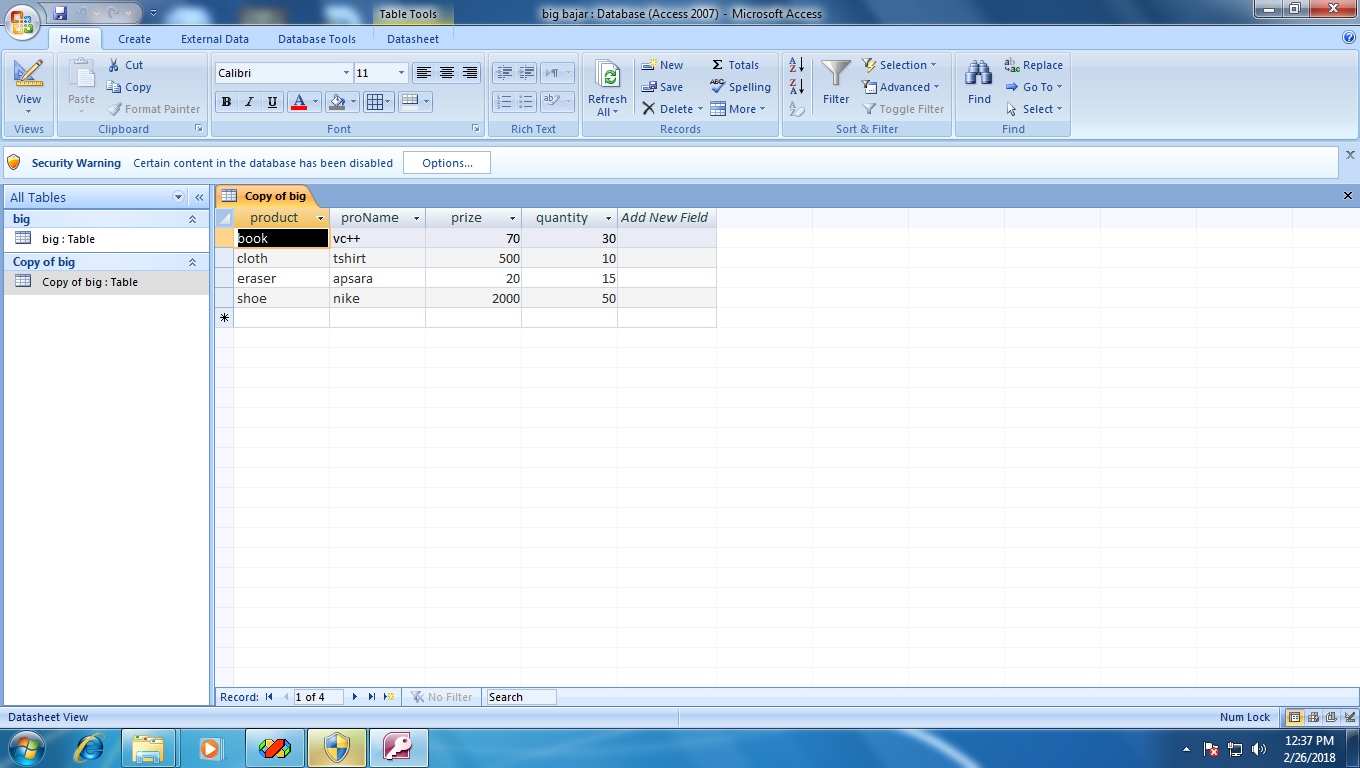
Original database



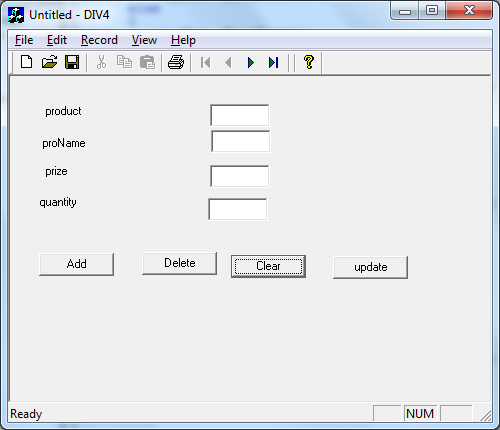
Add :-



**Delete :-**



Clear :-



Update :-

