

```

1  //////////////////////////////////////
2  //  First Win32 Program
3  //////////////////////////////////////
4
5  #include <windows.h>
6
7  LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
8
9  int WINAPI WinMain(
10      HINSTANCE hInstance,
11      HINSTANCE hPrevInstance,
12      LPSTR lpszCmdLine,
13      int iCmdShow
14  )
15  {
16      MSG msg;
17      BOOL bRet;
18      HWND hwnd;
19      WNDCLASSEX wndclass;
20
21      TCHAR szAppName[] = TEXT("MyWindow");
22
23      wndclass.cbSize = sizeof(wndclass);
24      wndclass.hInstance = hInstance;
25      wndclass.lpszClassName = szAppName;
26      wndclass.lpfnWndProc = WndProc;
27      wndclass.style = CS_HREDRAW | CS_VREDRAW;
28      wndclass.hIcon = LoadIcon(NULL, IDI_APPLICATION);
29      wndclass.hCursor = LoadCursor(NULL, IDC_ARROW);
30      wndclass.hbrBackground = (HBRUSH) GetStockObject(WHITE_BRUSH);
31      wndclass.lpszMenuName = NULL;
32      wndclass.cbClsExtra = 0;
33      wndclass.cbWndExtra = 0;
34      wndclass.hIconSm = LoadIcon(NULL, IDI_APPLICATION);
35
36      if(!RegisterClassEx(&wndclass))
37      {
38          MessageBox(NULL, TEXT("RegisterClassEx : FAILURE"), szAppName, 0);
39          return -1;
40      }
41
42      hwnd = CreateWindow(
43          szAppName,
44          TEXT("First Window"),
45          WS_OVERLAPPEDWINDOW,
46          CW_USEDEFAULT,
47          CW_USEDEFAULT,
48          CW_USEDEFAULT,
49          CW_USEDEFAULT,
50          NULL,
51          NULL,
52          hInstance,
53          NULL
54      );
55
56      if(hwnd == NULL)
57      {
58          MessageBox(NULL, TEXT("CreateWindow : FAILURE"), szAppName, 0);
59          return -1;
60      }
61
62      ShowWindow(hwnd, iCmdShow);
63      UpdateWindow(hwnd);
64
65
66
67

```

```

68     while((bRet = GetMessage(&msg, NULL, 0, 0)) != 0)
69     {
70         if(bRet == -1)
71         {
72             MessageBox(NULL, TEXT("CreateWindow : FAILURE"), szAppName, 0);
73             return -1;
74         }
75
76         TranslateMessage(&msg);
77         DispatchMessage(&msg);
78     }
79
80     return (int)msg.wParam;
81 }
82
83 LRESULT CALLBACK WndProc(
84     HWND hwnd,
85     UINT uiMessage,
86     WPARAM wParam,
87     LPARAM lParam
88 )
89 {
90     switch(uiMessage)
91     {
92     case WM_DESTROY:
93         PostQuitMessage(0);
94     }
95
96     return DefWindowProc(hwnd, uiMessage, wParam, lParam);
97 }
98

```