

```
1 // Headers
2 #include<windows.h>
3
4 // global function declarations
5 LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
6
7 // WinMain()
8 int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR  ↗
    lpzCmdLine, int iCmdShow)
9 {
10     // variable declarations
11     WNDCLASSEX wndclass;
12     HWND hwnd;
13     MSG msg;
14     TCHAR szAppName[] = TEXT("MyApp");
15
16     // code
17     // initialization of WNDCLASSEX
18     wndclass.cbSize = sizeof(WNDCLASSEX);
19     wndclass.style = CS_HREDRAW | CS_VREDRAW;
20     wndclass.cbClsExtra = 0;
21     wndclass.cbWndExtra = 0;
22     wndclass.lpfnWndProc = WndProc;
23     wndclass.hInstance = hInstance;
24     wndclass.hIcon = LoadIcon(NULL, IDI_APPLICATION);
25     wndclass.hCursor = LoadCursor(NULL, IDC_ARROW);
26     wndclass.hbrBackground = (HBRUSH)GetStockObject(BLACK_BRUSH);
27     wndclass.lpszClassName = szAppName;
28     wndclass.lpszMenuName = NULL;
29     wndclass.hIconSm = LoadIcon(NULL, IDI_APPLICATION);
30
31     // register above class
32     RegisterClassEx(&wndclass);
33
34     // create window
35     hwnd = CreateWindow(szAppName,
36         TEXT("My Application"),
37         WS_OVERLAPPEDWINDOW,
38         CW_USEDEFAULT,
39         CW_USEDEFAULT,
40         CW_USEDEFAULT,
41         CW_USEDEFAULT,
42         NULL,
43         NULL,
44         hInstance,
45         NULL);
46
47     ShowWindow(hwnd, iCmdShow);
48     UpdateWindow(hwnd);
49
50     // message loop
51     while (GetMessage(&msg, NULL, 0, 0))
52     {
53         TranslateMessage(&msg);
54         DispatchMessage(&msg);
55     }
```

```
56     return((int)msg.wParam);
57 }
58
59 LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM lParam)
60 {
61     // code
62     switch (iMsg)
63     {
64     case WM_DESTROY:
65         PostQuitMessage(0);
66         break;
67     }
68     return(DefWindowProc(hwnd, iMsg, wParam, lParam));
69 }
70
```