

```

1  #include <Windows.h>
2  #include <stdio.h>
3  #include <string.h>
4
5  #pragma comment(lib, "user32.lib")
6  #pragma comment(lib, "gdi32.lib")
7  #pragma comment(lib, "kernel32.lib")
8
9  /* GLOBAL DATA -> ARRAY OF struct StudentInfo */
10
11  typedef struct StudentInfo
12  {
13      INT iRollNumber;
14      TCHAR* szStudentName;
15      FLOAT fAttnd;
16      TCHAR* szCity;
17
18  }STUDENTINFO, *PSTUDENTINFO;
19
20  STUDENTINFO strec[] = {
21      {1, TEXT("SHUBHAM"), 90.5f, TEXT("Pune")},
22      {2, TEXT("ISHAN"), 89.2f, TEXT("Mumbai")},
23      {3, TEXT("SALONI"), 92.5f, TEXT("Pune")},
24      {4, TEXT("ANAGH"), 96.9f, TEXT("Pune")},
25      {5, TEXT("PRATIK"), 99.6f, TEXT("Jalgaon")},
26      {6, TEXT("TEJASWINI"), 91.1f, TEXT("Nanded")},
27      {7, TEXT("NIKITA"), 93.0f, TEXT("Thane")},
28      {8, TEXT("SHREYAS"), 80.3f, TEXT("Satara")},
29      {9, TEXT("ADITYA"), 87.2f, TEXT("Yavatmal")},
30      {10, TEXT("VEENA"), 82.2f, TEXT("Nanded")},
31      {11, TEXT("MAHENDRA"), 73.2f, TEXT("Solapur")},
32      {12, TEXT("MAYUR"), 93.7f, TEXT("Solapur")},
33      {13, TEXT("PAYAL"), 83.8f, TEXT("Kolhapur")},
34      {14, TEXT("SWATI"), 73.0f, TEXT("Pune")},
35      {15, TEXT("AKSHAY"), 55.1f, TEXT("Mumbai")},
36      {16, TEXT("KISHOR"), 28.3f, TEXT("Mumbai")},
37      {17, TEXT("SATEJ"), 91.9f, TEXT("Thane")},
38      {18, TEXT("SANDHYA"), 42.0f, TEXT("Thane")},
39      {19, TEXT("SOMESH"), 98.7f, TEXT("Thane")},
40      {20, TEXT("SHIVANI"), 51.2f, TEXT("Beed")},
41      {21, TEXT("GANESH"), 61.1f, TEXT("Nashik")},
42      {22, TEXT("VAISHNAVI"), 72.3f, TEXT("Nashik")},
43      {23, TEXT("JAI"), 97.0f, TEXT("Raigad")},
44      {24, TEXT("PARIKSHIT"), 48.4f, TEXT("Ratnagiri")},
45      {25, TEXT("MANGESH"), 66.7f, TEXT("Ratnagiri")},
46      {26, TEXT("RAKESH"), 87.9f, TEXT("Raigad")},
47      {27, TEXT("AARYA"), 77.1f, TEXT("Sindhudurga")},
48      {28, TEXT("ONKAR"), 59.4f, TEXT("Sindhudurga")},
49      {29, TEXT("RISHIKESH"), 63.8f, TEXT("Gadchiroli")},
50      {30, TEXT("MALLIKARJUN"), 58.8f, TEXT("Akole")},
51      {31, TEXT("REVAN"), 72.9f, TEXT("Akole")},
52      {32, TEXT("MADHURI"), 81.1f, TEXT("Gadchiroli")},
53      {33, TEXT("KARTIK"), 94.1f, TEXT("Sangli")},
54      {34, TEXT("UTKARSH"), 97.4f, TEXT("Sangli")},
55      {35, TEXT("VIVEK"), 89.3f, TEXT("Latur")},
56      {36, TEXT("NISHA"), 17.7f, TEXT("Hingoli")},
57      {37, TEXT("SHAHED"), 21.8f, TEXT("Hingoli")},
58      {38, TEXT("MINAL"), 78.8f, TEXT("Parbhani")},
59      {39, TEXT("PRATIK"), 99.8f, TEXT("Ahmednagar")},
60      {40, TEXT("SAMADHAN"), 89.1f, TEXT("Ahmednagar")},
61      {41, TEXT("YADNESH"), 82.2f, TEXT("Jalna")},
62      {42, TEXT("YASHASVI"), 95.2f, TEXT("Washim")},
63      {43, TEXT("PRATHAMESH"), 80.4f, TEXT("Buldhana")},
64      {44, TEXT("SUMIT"), 96.4f, TEXT("Wardha")},
65      {45, TEXT("PRAJWAL"), 93.6f, TEXT("Chandrapur")},
66      {46, TEXT("OM"), 67.5f, TEXT("Gondia")},
67      {47, TEXT("SUVARNA"), 87.9f, TEXT("Bhandara")},
68      {48, TEXT("SAKSHI"), 64.1f, TEXT("Dhule")},
69      {49, TEXT("VIKRAM"), 90.2f, TEXT("Nandurbar")},

```

```

70     {50, TEXT("BHAGWAN"), 57.2f, TEXT("Osmanabad")},
71     {51, TEXT("MANGESH"), 66.7f, TEXT("Ratnagiri")},
72     {52, TEXT("RAKESH"), 87.9f, TEXT("Raigad")},
73     {53, TEXT("AARYA"), 77.1f, TEXT("Sindhudurga")},
74     {54, TEXT("ONKAR"), 59.4f, TEXT("Sindhudurga")},
75     {55, TEXT("RISHIKESH"), 63.8f, TEXT("Gadchiroli")},
76     {56, TEXT("MALLIKARJUN"), 58.8f, TEXT("Akole")},
77     {57, TEXT("REVAN"), 72.9f, TEXT("Akole")},
78     {58, TEXT("MADHURI"), 81.1f, TEXT("Gadchiroli")},
79     {59, TEXT("KARTIK"), 94.1f, TEXT("Sangli")},
80     {60, TEXT("UTKARSH"), 97.4f, TEXT("Sangli")},
81     {61, TEXT("VIVEK"), 89.3f, TEXT("Latur")},
82     {62, TEXT("NISHA"), 17.7f, TEXT("Hingoli")},
83     {63, TEXT("SHAHED"), 21.8f, TEXT("Hingoli")},
84     {64, TEXT("MINAL"), 78.8f, TEXT("Parbhani")},
85     {65, TEXT("PRATIK"), 99.8f, TEXT("Ahmednagar")},
86     {66, TEXT("SAMADHAN"), 89.1f, TEXT("Ahmednagar")},
87     {67, TEXT("YADNESH"), 82.2f, TEXT("Jalna")},
88     {68, TEXT("YASHASVI"), 95.2f, TEXT("Washim")},
89     {69, TEXT("PRATHAMESH"), 80.4f, TEXT("Buldhana")},
90     {70, TEXT("SUMIT"), 96.4f, TEXT("Wardha")},
91     {71, TEXT("PRAJWAL"), 93.6f, TEXT("Chandrapur")},
92     {72, TEXT("OM"), 67.5f, TEXT("Gondia")},
93     {73, TEXT("SUVARNA"), 87.9f, TEXT("Bhandara")},
94     {74, TEXT("SAKSHI"), 64.1f, TEXT("Dhule")},
95     {75, TEXT("VIKRAM"), 90.2f, TEXT("Nandurbar")},
96 };
97
98 LRESULT CALLBACK WndProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam);
99
100 int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int
nShowCmd)
101 {
102     static TCHAR szClassName[] = TEXT("The Standard Window");
103     static TCHAR szTitleOfWindow[] = TEXT("STUDENT INFO");
104
105     WNDCLASSEX wnd;
106     MSG msg;
107
108     HWND hwnd = NULL;
109
110     ZeroMemory(&wnd, sizeof(WNDCLASSEX));
111     ZeroMemory(&msg, sizeof(MSG));
112
113     wnd.cbSize = sizeof(WNDCLASSEX);
114     wnd.cbClsExtra = 0;
115     wnd.cbWndExtra = 0;
116     wnd.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);
117     wnd.hCursor = LoadCursor(NULL, IDC_ARROW);
118     wnd.hIcon = LoadIcon(NULL, IDI_APPLICATION);
119     wnd.hIconSm = LoadIcon(NULL, IDI_APPLICATION);
120     wnd.hInstance = hInstance;
121     wnd.lpfnWndProc = WndProc;
122     wnd.lpszClassName = szClassName;
123     wnd.lpszMenuName = NULL;
124     wnd.style = CS_HREDRAW | CS_VREDRAW;
125
126     if(!RegisterClassEx(&wnd))
127     {
128         MessageBox(NULL, TEXT("Failed to register a window class"),
129             TEXT("RegisterClassEx"), MB_ICONERROR | MB_TOPMOST);
130         ExitProcess(EXIT_FAILURE);
131     }
132
133     hwnd = CreateWindowEx(WS_EX_APPWINDOW, szClassName, szTitleOfWindow,
WS_OVERLAPPEDWINDOW,
134         CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,
135         NULL, NULL, hInstance, NULL
136     );

```

```

137
138     if(hwnd == NULL)
139     {
140         MessageBox(NULL, TEXT("Failed to create window"), TEXT("CreateWindowEx"),
141             MB_ICONERROR | MB_TOPMOST);
142         ExitProcess(EXIT_FAILURE);
143     }
144
145     ShowWindow(hwnd, nShowCmd);
146     UpdateWindow(hwnd);
147
148     while(GetMessage(&msg, NULL, 0, 0))
149     {
150         TranslateMessage(&msg);
151         DispatchMessage(&msg);
152     }
153
154     return (msg.wParam);
155 }
156
157 LRESULT CALLBACK WndProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
158 {
159     static int cxChar, cyChar, cxCaps;
160     int i;
161     HDC hdc;
162     TEXTMETRIC tm;
163     PAINTSTRUCT ps;
164     TCHAR szRollBuffer[16];
165     TCHAR szAttndBuffer[16];
166     int iRollLen, iAttndLen;
167
168     switch(uMsg)
169     {
170     case WM_CREATE:
171         hdc = GetDC(hwnd);
172         GetTextMetrics(hdc, &tm);
173         cxChar = tm.tmAveCharWidth;
174         cxCaps = (tm.tmPitchAndFamily & 1 ? 3 : 2) * cxChar / 2;
175         cyChar = tm.tmHeight + tm.tmExternalLeading;
176         ReleaseDC(hwnd, hdc);
177         break;
178
179     case WM_PAINT:
180         hdc = BeginPaint(hwnd, &ps);
181         for(i = 0; i < sizeof(strec)/sizeof(strec[0]); ++i)
182         {
183             memset(szRollBuffer, 0, 16);
184             memset(szAttndBuffer, 0, 16);
185             sprintf(szRollBuffer, 16, "%d", strec[i].iRollNumber);
186             sprintf(szAttndBuffer, 16, "%.1f", strec[i].fAttnd);
187             iRollLen = strlen(szRollBuffer);
188             iAttndLen = strlen(szAttndBuffer);
189             TextOut(hdc, 0, i * cyChar, szRollBuffer, iRollLen);
190             TextOut(hdc, 5 * cxChar, i * cyChar, strec[i].szStudentName,
191                 strlen(strec[i].szStudentName)
192             );
193             TextOut(hdc, 5 * cxChar + 16 * cxCaps, i * cyChar, szAttndBuffer,
194                 iAttndLen);
195             TextOut(hdc, 20 * cxChar + 16 * cxCaps, i * cyChar, strec[i].szCity,
196                 strlen(strec[i].szCity)
197             );
198             EndPaint(hwnd, &ps);
199             break;
200
201     case WM_DESTROY:
202         PostQuitMessage(0);
203         break;
204     }

```

```
205
206     return (DefWindowProc(hwnd, uMsg, wParam, lParam));
207 }
```