```cpp
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <memory.h>

#include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <X11/XKBlib.h>
#include <X11/keysym.h>

//namespaces
using namespace std;

//global variable declarations
bool bFullscreen=false;
Display *gpDisplay=NULL;
XVisualInfo *gpXVisualInfo=NULL;
Colormap gColormap;
Window gWindow;
int giWindowWidth=800;
int giWindowHeight=600;

//entry-point function
int main(void)
{
    //function prototypes
    void CreateWindow(void);
    void ToggleFullscreen(void);
    void uninitialize();

    //variable declarations
    int winWidth=giWindowWidth;
    int winHeight=giWindowHeight;

    //code
    CreateWindow();

    //Message Loop
    XEvent event;
    KeySym keysym;

    while(1)
    {
        XNextEvent(gpDisplay,&event);
        switch(event.type)
        {
            case MapNotify:
                break;
            case KeyPress:
                keysym=XkbKeycodeToKeysym(gpDisplay,event.xkey.keycode,0,0);
                switch(keysym)
                {
                    case XK_Escape:
```

```
54                          uninitialize();
55                          exit(0);
56                      case XK_F:
57                      case XK_f:
58                          if(bFullscreen==false)
59                          {
60                              ToggleFullscreen();
61                              bFullscreen=true;
62                          }
63                          else
64                          {
65                              ToggleFullscreen();
66                              bFullscreen=false;
67                          }
68                          break;
69                      default:
70                          break;
71                  }
72                  break;
73              case ButtonPress:
74                  switch(event.xbutton.button)
75                  {
76                      case 1:
77                          break;
78                      case 2:
79                          break;
80                      case 3:
81                          break;
82                      default:
83                          break;
84                  }
85                  break;
86              case MotionNotify:
87                  break;
88              case ConfigureNotify:
89                  winWidth=event.xconfigure.width;
90                  winHeight=event.xconfigure.height;
91                  break;
92              case Expose:
93                  break;
94              case DestroyNotify:
95                  break;
96              case 33:
97                  uninitialize();
98                  exit(0);
99              default:
100                 break;
101         }
102     }
103
104     uninitialize();
105     return(0);
106 }
```

```
107
108 void CreateWindow(void)
109 {
110     //function prorttypes
111     void uninitialize(void);
112
113     //variable declarations
114     XSetWindowAttributes winAttribs;
115     int defaultScreen;
116     int defaultDepth;
117     int styleMask;
118
119     //code
120     gpDisplay=XOpenDisplay(NULL);
121     if(gpDisplay==NULL)
122     {
123         printf("ERROR : Unable To Open X Display.\nExitting Now...\n");
124         uninitialize();
125         exit(1);
126     }
127
128     defaultScreen=XDefaultScreen(gpDisplay);
129
130     defaultDepth=DefaultDepth(gpDisplay,defaultScreen);
131
132     gpXVisualInfo=(XVisualInfo *)malloc(sizeof(XVisualInfo));
133     if(gpXVisualInfo==NULL)
134     {
135         printf("ERROR : Unable To Allocate Memory For Visual Info.\nExitting
            Now...\n");
136         uninitialize();
137         exit(1);
138     }
139
140     XMatchVisualInfo
          (gpDisplay,defaultScreen,defaultDepth,TrueColor,gpXVisualInfo);
141     if(gpXVisualInfo==NULL)
142     {
143         printf("ERROR : Unable To Get A Visual.\nExitting Now...\n");
144         uninitialize();
145         exit(1);
146     }
147
148     winAttribs.border_pixel=0;
149     winAttribs.background_pixmap=0;
150     winAttribs.colormap=XCreateColormap(gpDisplay,
151                         RootWindow(gpDisplay, gpXVisualInfo->screen),
152                         gpXVisualInfo->visual,
153                         AllocNone);
154     gColormap=winAttribs.colormap;
155
156     winAttribs.background_pixel=BlackPixel(gpDisplay,defaultScreen);
157
```

```
158     winAttribs.event_mask= ExposureMask | VisibilityChangeMask |
          ButtonPressMask | KeyPressMask | PointerMotionMask |
159                   StructureNotifyMask;
160
161     styleMask=CWBorderPixel | CWBackPixel | CWEventMask | CWColormap;
162
163     gWindow=XCreateWindow(gpDisplay,
164                   RootWindow(gpDisplay,gpXVisualInfo->screen),
165                   0,
166                   0,
167                   giWindowWidth,
168                   giWindowHeight,
169                   0,
170                   gpXVisualInfo->depth,
171                   InputOutput,
172                   gpXVisualInfo->visual,
173                   styleMask,
174                   &winAttribs);
175     if(!gWindow)
176     {
177         printf("ERROR : Failed To Create Main Window.\nExitting Now...\n");
178         uninitialize();
179         exit(1);
180     }
181
182     XStoreName(gpDisplay,gWindow,"First XWindow");
183
184     Atom windowManagerDelete=XInternAtom(gpDisplay,"WM_DELETE_WINDOW",True);
185     XSetWMProtocols(gpDisplay,gWindow,&windowManagerDelete,1);
186
187     XMapWindow(gpDisplay,gWindow);
188 }
189
190 void ToggleFullscreen(void)
191 {
192     //variable declarations
193     Atom wm_state;
194     Atom fullscreen;
195     XEvent xev={0};
196
197     //code
198     wm_state=XInternAtom(gpDisplay,"_NET_WM_STATE",False);
199     memset(&xev,0,sizeof(xev));
200
201     xev.type=ClientMessage;
202     xev.xclient.window=gWindow;
203     xev.xclient.message_type=wm_state;
204     xev.xclient.format=32;
205     xev.xclient.data.l[0]=bFullscreen ? 0 : 1;
206
207     fullscreen=XInternAtom(gpDisplay,"_NET_WM_STATE_FULLSCREEN",False);
208     xev.xclient.data.l[1]=fullscreen;
209
```

```
210     XSendEvent(gpDisplay,
211                RootWindow(gpDisplay,gpXVisualInfo->screen),
212                False,
213                StructureNotifyMask,
214                &xev);
215 }
216
217 void uninitialize(void)
218 {
219     if(gWindow)
220     {
221         XDestroyWindow(gpDisplay,gWindow);
222     }
223
224     if(gColormap)
225     {
226         XFreeColormap(gpDisplay,gColormap);
227     }
228
229     if(gpXVisualInfo)
230     {
231         free(gpXVisualInfo);
232         gpXVisualInfo=NULL;
233     }
234
235     if(gpDisplay)
236     {
237         XCloseDisplay(gpDisplay);
238         gpDisplay=NULL;
239     }
240 }
241
242
```