

```

1  #define UNICODE
2  #include<windows.h>
3  #include"ClassFactoryDllServerWithRegFile.h"
4  // global function declarations
5  LRESULT CALLBACK WndProc(HWND,UINT,WPARAM,LPARAM);
6  // global variable declarations
7  ISum *pISum=NULL;
8  ISubtract *pISubtract=NULL;
9  // WinMain
10 int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,
11                    LPSTR lpCmdLine,int nCmdShow)
12 {
13     // variable declarations
14     WNDCLASSEX wndclass;
15     HWND hwnd;
16     MSG msg;
17     TCHAR AppName[]=TEXT("ComClient");
18     HRESULT hr;
19     // code
20     // COM Initialization
21     hr=CoInitialize(NULL);
22     if(FAILED(hr))
23     {
24         MessageBox(NULL,TEXT("COM Library Can Not Be Initialized.\nProgram Will
25             Now Exit."),TEXT("Program Error"),MB_OK);
26         exit(0);
27     }
28     // WNDCLASSEX initialization
29     wndclass.cbSize=sizeof(wndclass);
30     wndclass.style=CS_HREDRAW|CS_VREDRAW;
31     wndclass.cbClsExtra=0;
32     wndclass.cbWndExtra=0;
33     wndclass.lpfnWndProc=WndProc;
34     wndclass.hIcon=LoadIcon(NULL,IDI_APPLICATION);
35     wndclass.hCursor=LoadCursor(NULL,IDC_ARROW);
36     wndclass.hbrBackground=(HBRUSH)GetStockObject(WHITE_BRUSH);
37     wndclass.hInstance=hInstance;
38     wndclass.lpszClassName=AppName;
39     wndclass.lpszMenuName=NULL;
40     wndclass.hIconSm=LoadIcon(NULL,IDI_APPLICATION);
41     // register window class
42     RegisterClassEx(&wndclass);
43     // create window
44     hwnd=CreateWindow(AppName,
45                       TEXT("Client Of COM Dll Server"),
46                       WS_OVERLAPPEDWINDOW,
47                       CW_USEDEFAULT,
48                       CW_USEDEFAULT,
49                       CW_USEDEFAULT,
50                       CW_USEDEFAULT,
51                       NULL,

```

```

52         hInstance,
53         NULL);
54     ShowWindow(hwnd,nCmdShow);
55     UpdateWindow(hwnd);
56     // message loop
57     while(GetMessage(&msg,NULL,0,0))
58     {
59         TranslateMessage(&msg);
60         DispatchMessage(&msg);
61     }
62     // COM Un-initialization
63     CoUninitialize();
64     return((int)msg.wParam);
65 }
66 // Window Procedure
67 LRESULT CALLBACK WndProc(HWND hwnd,UINT iMsg,WPARAM wParam,LPARAM lParam)
68 {
69     // function declarations
70     void SafeInterfaceRelease(void);
71     // variable declarations
72     HRESULT hr;
73     int iNum1,iNum2,iSum;
74     TCHAR str[255];
75     // code
76     switch(iMsg)
77     {
78     case WM_CREATE:
79         hr=CoCreateInstance(CLSID_SumSubtract,NULL,CLSCTX_INPROC_SERVER,
80             IID_ISum,(void **)&pISum);
81         if(FAILED(hr))
82         {
83             MessageBox(hwnd,TEXT("ISum Interface Can Not Be Obtained"),TEXT
84                 ("Error"),MB_OK);
85             DestroyWindow(hwnd);
86         }
87         // initialize arguments hardcoded
88         iNum1=55;
89         iNum2=45;
90         // call SumOfTwoIntegers() of ISum to get the sum
91         pISum->SumOfTwoIntegers(iNum1,iNum2,&iSum);
92         // display the result
93         wsprintf(str,TEXT("Sum Of %d And %d = %d"),iNum1,iNum2,iSum);
94         MessageBox(hwnd,str,TEXT("Result"),MB_OK);
95         // call QueryInterface() on ISum,to get ISubtract's pointer
96         hr=pISum->QueryInterface(IID_ISubtract,(void **)&pISubtract);
97         if(FAILED(hr))
98         {
99             MessageBox(hwnd,TEXT("ISubtract Interface Can Not Be Obtained"),TEXT
100                 ("Error"),MB_OK);
101             DestroyWindow(hwnd);
102         }
103         // as ISum is now not needed onwords, release it

```

```
102     pISum->Release();
103     pISum=NULL;// make released interface NULL
104     // again initialize arguments hardcoded
105     iNum1=155;
106     iNum2=145;
107     // again call SumOfTwoIntegers() of ISum to get the new sum
108     pISubtract->SubtractionOfTwoIntegers(iNum1,iNum2,&iSum);
109     // as ISum is now not needed onwords, release it
110     pISubtract->Release();
111     pISubtract=NULL;// make released interface NULL
112     // display the result
113     wsprintf(str,TEXT("Subtraction Of %d And %d = %d"),iNum1,iNum2,iSum);
114     MessageBox(hwnd,str,TEXT("Result"),MB_OK);
115     // exit the application
116     DestroyWindow(hwnd);
117     break;
118 case WM_DESTROY:
119     PostQuitMessage(0);
120     break;
121 }
122 return(DefWindowProc(hwnd,iMsg,wParam,lParam));
123 }
124 void SafeInterfaceRelease(void)
125 {
126     // code
127     if(pISum)
128     {
129         pISum->Release();
130         pISum=NULL;
131     }
132     if(pISubtract)
133     {
134         pISubtract->Release();
135         pISubtract=NULL;
136     }
137 }
138
```