UCS638 Assignment: PE Code Injection
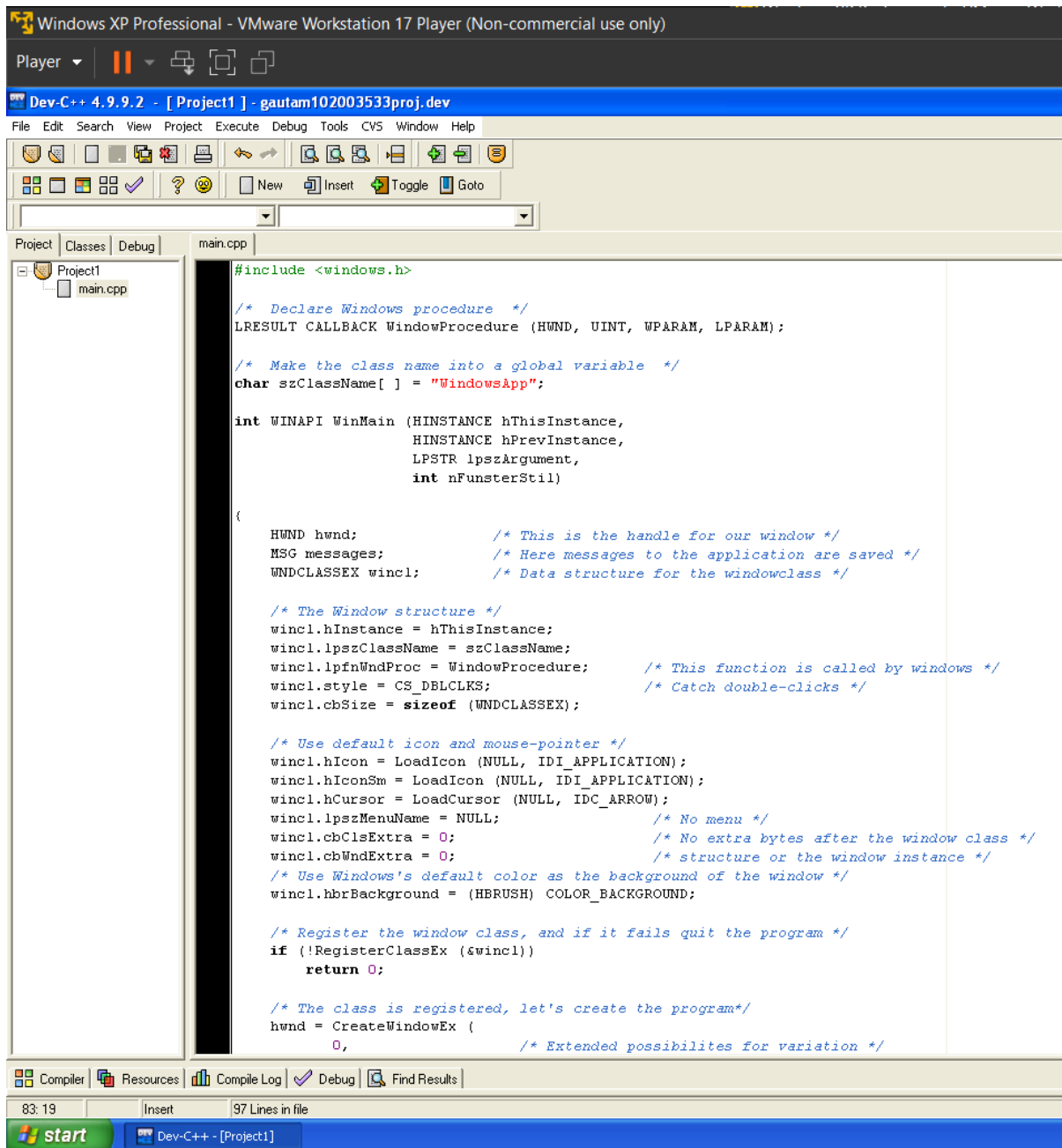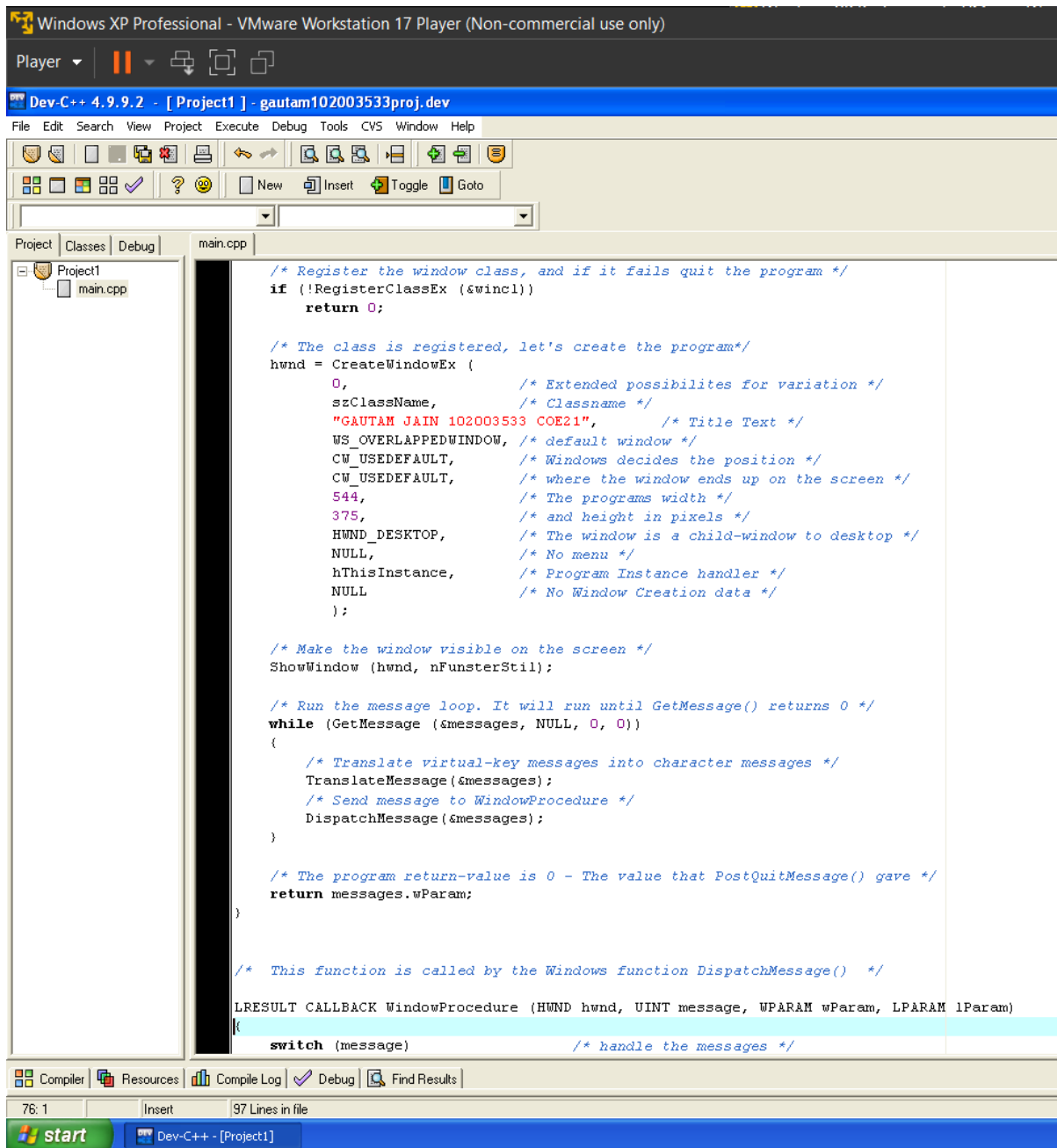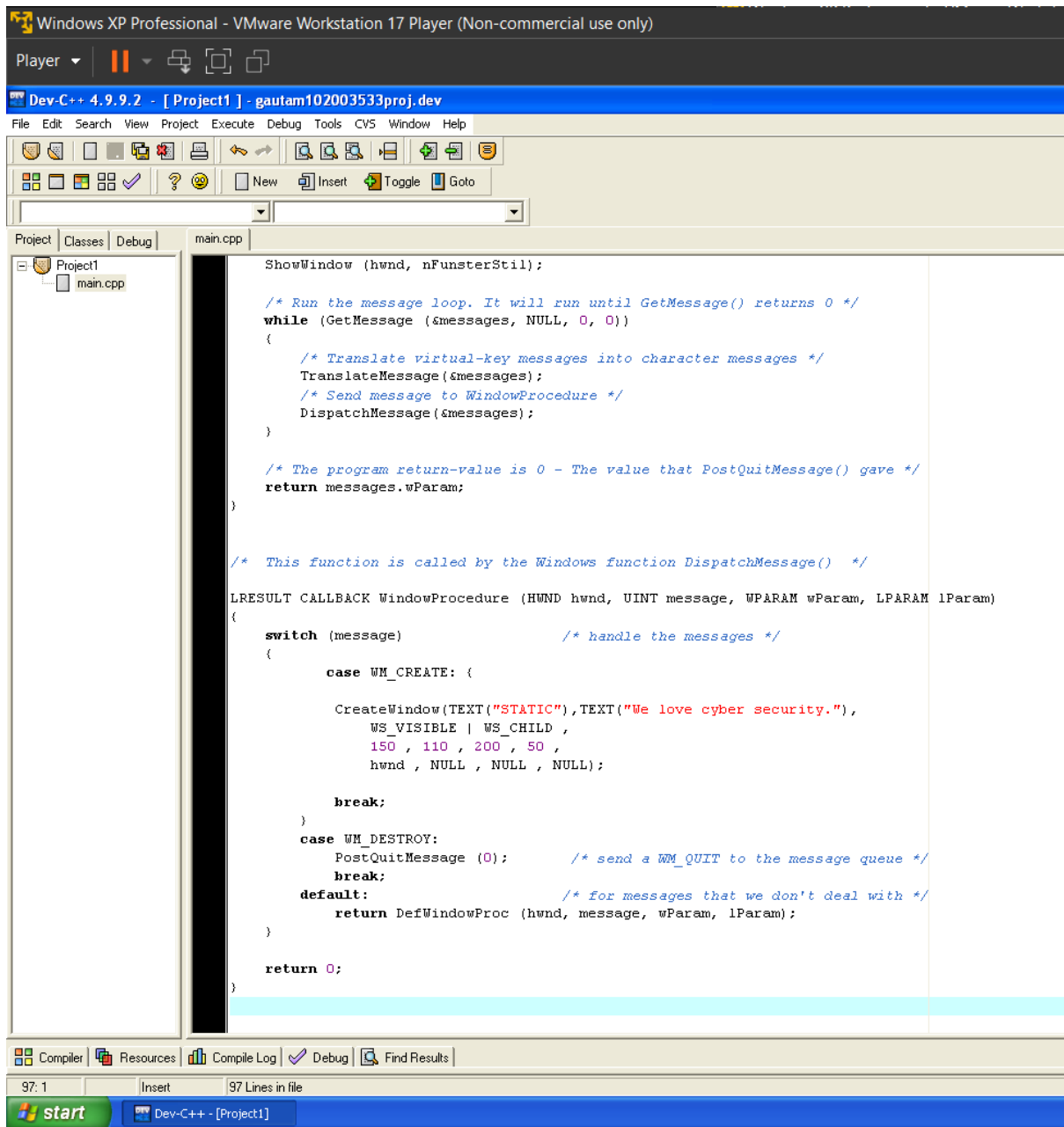
Submitted by Gautam Jain, 102003533, COE21

Creating Windows Application by writing code in DevC++ IDE.

Dev-C++ 4.9.9.2 - [Project1] - gautam102003533proj.dev

File   Edit   Search   View   Project   Execute   Debug   Tools   CVS   Window   Help

New   Insert   Toggle   Goto

Project | Classes | Debug   main.cpp

Project1
    main.cpp

```cpp
        /* Register the window class, and if it fails quit the program */
        if (!RegisterClassEx (&wincl))
            return 0;

        /* The class is registered, let's create the program*/
        hwnd = CreateWindowEx (
               0,                     /* Extended possibilites for variation */
               szClassName,           /* Classname */
               "GAUTAM JAIN 102003533 COE21",      /* Title Text */
               WS_OVERLAPPEDWINDOW,   /* default window */
               CW_USEDEFAULT,         /* Windows decides the position */
               CW_USEDEFAULT,         /* where the window ends up on the screen */
               544,                   /* The programs width */
               375,                   /* and height in pixels */
               HWND_DESKTOP,          /* The window is a child-window to desktop */
               NULL,                  /* No menu */
               hThisInstance,         /* Program Instance handler */
               NULL                   /* No Window Creation data */
               );

        /* Make the window visible on the screen */
        ShowWindow (hwnd, nFunsterStil);

        /* Run the message loop. It will run until GetMessage() returns 0 */
        while (GetMessage (&messages, NULL, 0, 0))
        {
            /* Translate virtual-key messages into character messages */
            TranslateMessage(&messages);
            /* Send message to WindowProcedure */
            DispatchMessage(&messages);
        }

        /* The program return-value is 0 - The value that PostQuitMessage() gave */
        return messages.wParam;
}


/*  This function is called by the Windows function DispatchMessage()  */

LRESULT CALLBACK WindowProcedure (HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
{
        switch (message)                  /* handle the messages */
```

Compiler | Resources | Compile Log | Debug | Find Results

76: 1   Insert   97 Lines in file

start   Dev-C++ - [Project1]

Dev-C++ 4.9.9.2 - [Project1] - gautam102003533proj.dev

File   Edit   Search   View   Project   Execute   Debug   Tools   CVS   Window   Help

Project | Classes | Debug   main.cpp

□ Project1
   main.cpp

```
        ShowWindow (hwnd, nFunsterStil);

        /* Run the message loop. It will run until GetMessage() returns 0 */
        while (GetMessage (&messages, NULL, 0, 0))
        {
            /* Translate virtual-key messages into character messages */
            TranslateMessage(&messages);
            /* Send message to WindowProcedure */
            DispatchMessage(&messages);
        }

        /* The program return-value is 0 - The value that PostQuitMessage() gave */
        return messages.wParam;
}


/*  This function is called by the Windows function DispatchMessage()  */

LRESULT CALLBACK WindowProcedure (HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)                    /* handle the messages */
    {
        case WM_CREATE: {

            CreateWindow(TEXT("STATIC"),TEXT("We love cyber security."),
                WS_VISIBLE | WS_CHILD ,
                150 , 110 , 200 , 50 ,
                hwnd , NULL , NULL , NULL);

            break;
        }
        case WM_DESTROY:
            PostQuitMessage (0);        /* send a WM_QUIT to the message queue */
            break;
        default:                        /* for messages that we don't deal with */
            return DefWindowProc (hwnd, message, wParam, lParam);
    }

    return 0;
}
```

Compiler | Resources | Compile Log | Debug | Find Results

97: 1   Insert   97 Lines in file

start   Dev-C++ - [Project1]

P.T.O.

32 bit executable:

GAUTAM JAIN 102003533 COE21

We love cyber security.