Implementing `Player`

Now that we've reversed the `Player` class, let's write our own program that makes a `Player` class and uses the functions related to the class.

```cpp
1   #include <iostream>
2   #include <Windows.h>
3
4   class Player {
5   public:
6       int score;
7       float health;
8       std::string name;
9   };
10
11  //void __cdecl InitializePlayer(class Player * __ptr64)
12  typedef void(WINAPI* IInitializePlayer)(Player*); // ?InitializePlayer@@YAXPEAVPlayer@@@Z
13  //void PrintPlayerStats(Player);
14  typedef void(WINAPI* IPrintPlayerStats)(Player); // PrintPlayerStats
15
16  int main()
17  {
18      Player player;
19      HMODULE dll = LoadLibraryA("DLL.DLL"); //Load our DLL.
20      if (dll != NULL)
21      {
22          //Initialize Player
23          IInitializePlayer InitializePlayer = (IInitializePlayer)GetProcAddress(dll, "?InitializePlayer@@YAXPEAVPlayer@@@Z");
24          if (InitializePlayer != NULL) {
25              InitializePlayer(&player);
26          }
27          else { printf("Can't load the function."); }
28
29          //PrintPlayerStats
30          IPrintPlayerStats PrintPlayerStats = (IPrintPlayerStats)GetProcAddress(dll, "PrintPlayerStats");
31          if (InitializePlayer != NULL) {
32              PrintPlayerStats(player);
33          }
34          else { printf("Can't load the function."); }
35
36      }
37  }
```

The player's name doesn't have to be a `std::string`, it can probably be a `const* char` as well.

Copy/Paste Code

```cpp
#include <iostream>
#include <Windows.h>

class Player {
public:
        int score;
        float health;
        std::string name;
};

//void __cdecl InitializePlayer(class Player * __ptr64)
typedef void(WINAPI* IInitializePlayer)(Player*); // ?
InitializePlayer@@YAXPEAVPlayer@@@Z
//void PrintPlayerStats(Player);
typedef void(WINAPI* IPrintPlayerStats)(Player); // PrintPlayerStats
```

```c
int main()
{
        Player player;
        HMODULE dll = LoadLibraryA("DLL.DLL"); //Load our DLL.
        if (dll != NULL)
        {
                //Initialize Player
                IInitializePlayer InitializePlayer =
(IInitializePlayer)GetProcAddress(dll, "?InitializePlayer@@YAXPEAVPlayer@@@Z");
                if (InitializePlayer != NULL) {
                        InitializePlayer(&player);
                }
                else { printf("Can't load the function."); }

                //PrintPlayerStats
                IPrintPlayerStats PrintPlayerStats =
(IPrintPlayerStats)GetProcAddress(dll, "PrintPlayerStats");
                if (InitializePlayer != NULL) {
                        PrintPlayerStats(player);
                }
                else { printf("Can't load the function."); }

        }
}
```