

SWINBURNE UNIVERSITY OF TECHNOLOGY

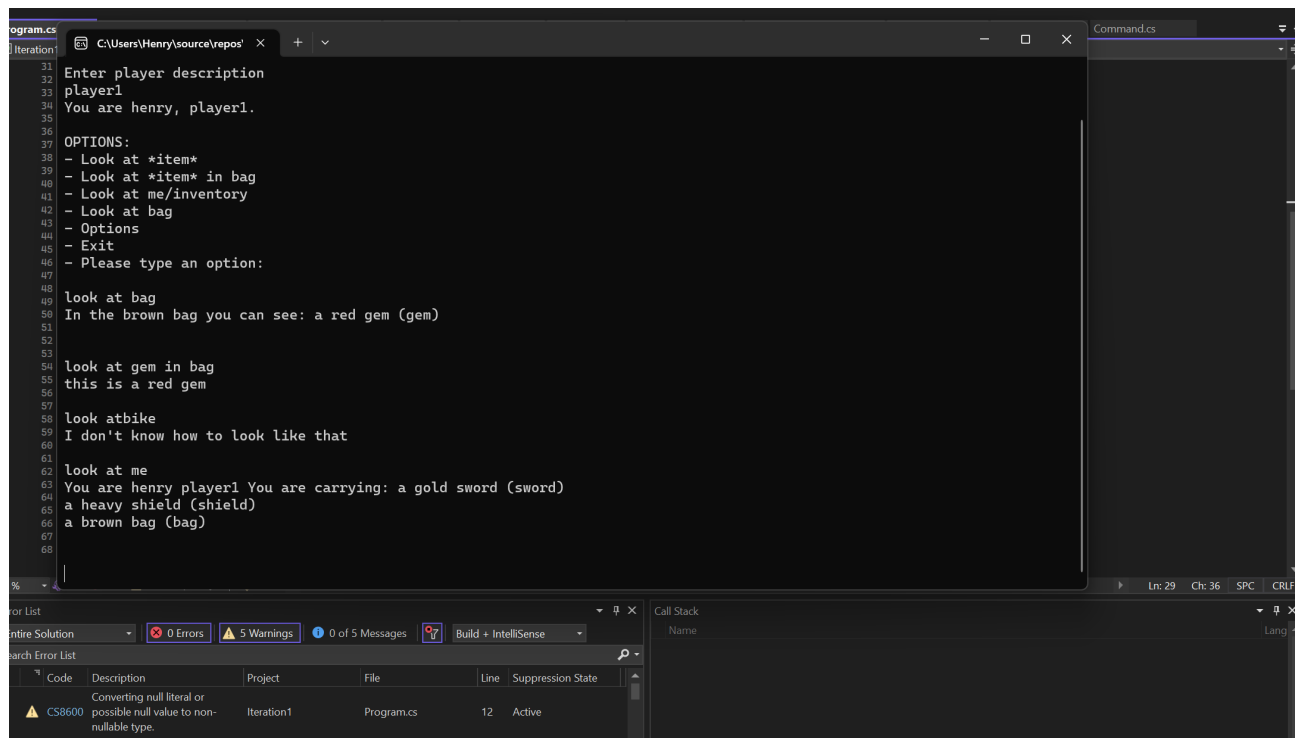
COS20007 OBJECT ORIENTED PROGRAMMING

7.1P - Case Study - Iteration 5 - Tying it Together

PDF generated at 10:37 on Monday 8th May, 2023

```
1  using System;
2
3  namespace Iteration1
4  {
5      public class Program
6      {
7
8          static void Main()
9          {
10
11              Console.WriteLine("Enter player name:");
12              string name = Console.ReadLine();
13
14              Console.WriteLine("\nEnter player description");
15              string desc = Console.ReadLine();
16
17              Console.WriteLine($"You are {name}, {desc}.");
18
19              Player player = new Player(name, desc);
20
21              Item shield = new Item(new string[] { "Shield" }, "heavy shield", "this is
↪ a heavy shield");
22              Item sword = new Item(new string[] { "Sword" }, "gold sword", "this is a
↪ gold sword");
23              Bag bag = new Bag(new string[] { "Bag" }, "brown bag", "this is a brown
↪ bag");
24              Item gem = new Item(new string[] { "Gem" }, "red gem", "this is a red
↪ gem");
25
26              player.Inventory.Put(sword);
27              player.Inventory.Put(shield);
28              player.Inventory.Put(bag);
29              bag.Inventory.Put(gem);
30
31              string options = ("\nOPTIONS:" +
32                  "\n- Look at *item*" +
33                  "\n- Look at *item* in bag" +
34                  "\n- Look at me/inventory" +
35                  "\n- Look at bag" +
36                  "\n- Options" +
37                  "\n- Exit" +
38                  "\n- Please type an option:\n");
39              Console.WriteLine(options);
40
41              string input = "";
42
43              while (input != "exit")
44              {
45                  input = Console.ReadLine().ToLower();
46                  string[] choice = input.Split(" ");
47
48                  Look_Command look;
```

```
50         look = new Look_Command();
51
52
53         if (input == "exit")
54         {
55             Console.WriteLine("\nGame Over.");
56         }
57
58         if (input == "options")
59         {
60             Console.WriteLine(options);
61         }
62
63         Console.WriteLine(look.Execute(player, choice)+ "\n");
64     }
65
66 }
67
68 }
```



The screenshot shows a Visual Studio IDE with a C# program and its output. The program is named `Program.cs` and is located at `C:\Users\Henry\source\repos\...`. The output window shows the following text:

```
Enter player description
player1
You are henry, player1.

OPTIONS:
- Look at *item*
- Look at *item* in bag
- Look at me/inventory
- Look at bag
- Options
- Exit
- Please type an option:

look at bag
In the brown bag you can see: a red gem (gem)

look at gem in bag
this is a red gem

look atbike
I don't know how to look like that

look at me
You are henry player1 You are carrying: a gold sword (sword)
a heavy shield (shield)
a brown bag (bag)
```

The bottom of the IDE shows the Error List and Call Stack. The Error List shows a warning (CS8600) at line 12 of `Program.cs`, which is "Converting null literal or possible null value to non-nullable type." The Call Stack is empty.