

SWINBURNE UNIVERSITY OF TECHNOLOGY

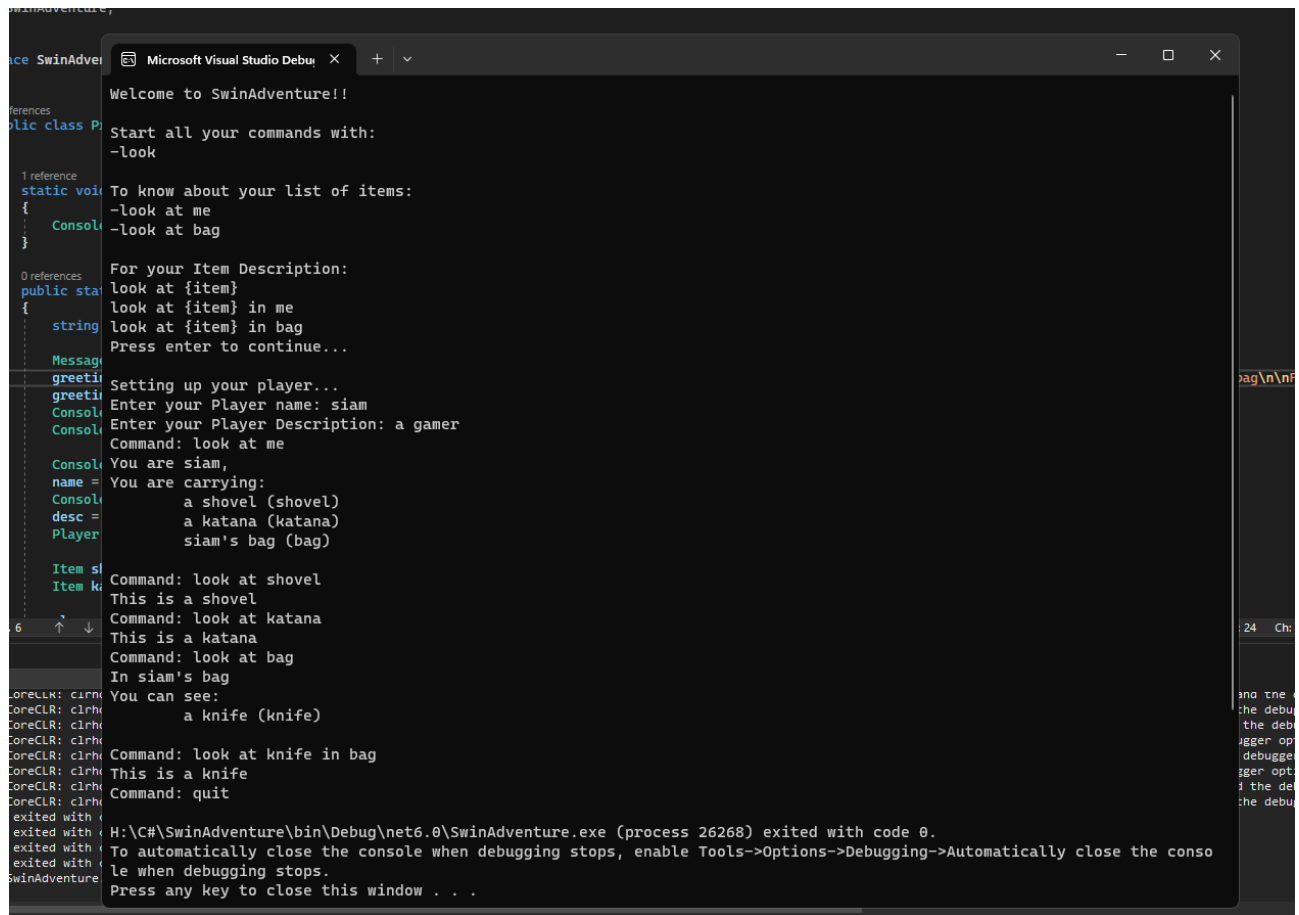
COS20007 OBJECT ORIENTED PROGRAMMING

Case Study - Iteration 5 - Tying it Together

PDF generated at 17:11 on Thursday 21st September, 2023

```
1  using System;
2  using System.Collections.Generic;
3  using System.Collections;
4  using System.Xml.Linq;
5  using SwinAdventure;
6
7
8  namespace SwinAdventure
9  {
10
11     public class Program
12     {
13
14         static void LookCommandExecution(Command command, string input, Player
↵ player)
15         {
16             Console.WriteLine(command.Execute(player, input.Split()));
17         }
18
19         public static void Main(string[] args)
20         {
21             string name, desc;
22
23             Message greetings;
24             greetings = new Message("Welcome to SwinAdventure!!\n\nStart all your
↵ commands with:\n-look\n\nTo know about your list of items:\n-look at me\n-look at
↵ bag\n\nFor your Item Description:\nlook at {item}\nlook at {item} in me\nlook at
↵ {item} in bag");
25             greetings.Print();
26             Console.WriteLine("Press enter to continue...");
27             Console.ReadLine();
28
29             Console.Write("Setting up your player...\nEnter your Player name: ");
30             name = Console.ReadLine();
31             Console.Write("Enter your Player Description: ");
32             desc = Console.ReadLine();
33             Player player = new Player(name, desc);
34
35             Item shovel = new Item(new string[] { "shovel" }, "a shovel", "This is a
↵ shovel");
36             Item katana = new Item(new string[] { "katana" }, "a katana", "This is a
↵ katana");
37
38             player.Inventory.Put(shovel);
39             player.Inventory.Put(katana);
40
41             Bag bag = new Bag(new string[] { $"bag" }, $"{player.Name}'s bag", $"This
↵ is {player.Name}'s bag");
42             Item knife = new Item(new string[] { "knife" }, "a knife", "This is a
↵ knife");
43
44             player.Inventory.Put(bag);
45             bag.Inventory.Put(knife);
```

```
46
47     string input;
48     Command command = new LookCommand();
49
50     while (true)
51     {
52         Console.Write("Command: ");
53         input = Console.ReadLine();
54         if (input == "quit")
55         {
56             break;
57         }
58         else
59         {
60             LookCommandExecution(command, input, player);
61         }
62     }
63
64 }
65
66
67
68 public class Message
69 {
70     private string _text;
71
72     public Message(string text)
73     {
74         _text = text;
75     }
76
77     public void Print()
78     {
79         Console.WriteLine(_text);
80     }
81 }
82
83
84 }
85 }
86
```



The screenshot shows the Microsoft Visual Studio Debug Console window for a project named 'SwinAdventure'. The console displays the output of a game execution. The game starts with a welcome message and instructions. The player enters 'siam' as their name and 'a gamer' as their description. The game then prompts the player to look at items. The player enters 'look at me', 'look at shovel', 'look at katana', 'look at bag', and 'look at knife in bag'. The game responds with the item descriptions. Finally, the player enters 'quit', and the game exits with code 0. The console also shows the file explorer on the left and the solution explorer on the right.

```
Microsoft Visual Studio Debug Console
Welcome to SwinAdventure!!
Start all your commands with:
-look

To know about your list of items:
-look at me
-look at bag

For your Item Description:
look at {item}
look at {item} in me
look at {item} in bag
Press enter to continue...

Setting up your player...
Enter your Player name: siam
Enter your Player Description: a gamer
Command: look at me
You are siam,
You are carrying:
    a shovel (shovel)
    a katana (katana)
    siam's bag (bag)

Command: look at shovel
This is a shovel
Command: look at katana
This is a katana
Command: look at bag
In siam's bag
You can see:
    a knife (knife)

Command: look at knife in bag
This is a knife
Command: quit

H:\C#\SwinAdventure\bin\Debug\net6.0\SwinAdventure.exe (process 26268) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```