

```

...10.1C\SwinAdventure\SwinAdventure\CommandProcessor.cs 1
1  //-The CommandProcessor class has to inherit from Command so far ➤
   because:-//
2  //-Firstly, It can be treated like any other command-//
3  //-Secondly, it has Execute method to process and delegate user ➤
   commands.-//
4  //-Lastly, It provides a consistent and extensible way to handle all game ➤
   commands.-//
5  using System;
6  using System.Collections.Generic;
7  using System.Diagnostics.Contracts;
8  using System.Linq;
9  using System.Text;
10 using System.Threading.Tasks;
11 using SwinAdventure;
12
13 namespace SwinAdventure
14 {
15     public class CommandProcessor : Command
16     {
17         //-Attributes-//
18         List<Command> _commands;
19         //-Constructor-//
20         public CommandProcessor() : base(new string[] {"command"})
21         {
22             _commands = new List<Command>();
23             _commands.Add(new LookCommand());
24             _commands.Add(new MoveCommand());
25         }
26         //-Methods-//
27         public override string Execute(Player p, string[] text)
28         {
29             if (text.Length == 0)
30             {
31                 return "Please enter a command.";
32             }
33             else if (text.Length > 1)
34             {
35                 string commandId = text[0].ToLower();
36
37                 foreach (Command cmd in _commands)
38                 {
39                     if (cmd.AreYou(commandId))
40                     {
41                         return cmd.Execute(p, text);
42                     }
43                 }
44             }
45             return "There is no command like that."; // Or "I don't know
               that command." based on your preference ➤

```

```
46         }  
47     }  
48 }  
49
```