```
1  using System;
2  using System.ComponentModel.Design; // This using directive is not strictly necessary for the provided code, but kept as it was in the original.
3
4  namespace SwinAdventure
5  {
6      internal class Program
7      {
8          static void Main(string[] args)
9          {
10             // Print sentences to the console window
11             Console.WriteLine("WELCOME TO THE SWINADVENTURE GAME!");
12             Console.WriteLine("-----------------------------------");
13             // Enter player name and player description
14             Console.Write("Dear Warrior, please enter your name: ");
15             string playername = Console.ReadLine();
16             Console.Write("Please enter your description: "); // Changed to Write for better flow
17             string playerdescription = Console.ReadLine();
18             Player player = new Player(playername, playerdescription);
19             // Set up initial items
20             Item axe = new Item(new string[] { "sword" }, "a sharp sword", "+20 ATK points"); // Added "axe" alias
21             Item shield = new Item(new string[] { "shield" }, "a bronze shield", "+5 DEF points ");
22             Bag backpack = new Bag(new string[] { "backpack"}, "A heavy backpack", "Contains crucial items"); // Added "backpack" alias
23             // Put items into the player's inventory
24             player.Inventory.Put(axe);
25             player.Inventory.Put(shield);
26             player.Inventory.Put(backpack);
27             // Create items and put them into the backpack
28             Item gun = new Item(new string[] { "gun"}, "An AK-47 gun", "+15 ATK points"); // Added "ak47" alias
29             backpack.Inventory.Put(gun);
30             Item shovel = new Item(new string[] { "shovel" }, "An useful shovel", "+10 ATK points");
31             backpack.Inventory.Put(shovel);
32             Item map = new Item(new string[] { "map" }, "A detailed map", "Used for showing directions");
33             backpack.Inventory.Put(map);
34             Item book = new Item(new string[] { "book"}, "A thick book", "Contains knowledge of human"); // Added "tome" alias
35             backpack.Inventory.Put(book);
36             //Paths, Directions, and Locations
37             Location university = new Location("An old university", "A mystery university");
```

```
38              Location library = new Location("State Library", "An old and
                  unforgettable library");
39              Location mainhall = new Location("School Mainhall", "A large
                  mainhall of a university");
40              Location principalroom = new Location("Principal Room", "The
                  pricipal room");
41              Location pavement = new Location("A pavement", "Outside of
                  the university");
42              Location street = new Location("A street", "The Alizabeth
                  street");
43              Paths pavementtoMainHall = new Paths(new string[] { "
                  forward", "north" }, "forward path", "The way to the Main
                  Hall of the University", mainhall);
44              Paths mainhalltopavement = new Paths(new string[]
                  { "backward", "south" }, "backward path", "The way back to
                  the pavement", pavement);
45              Paths mainhalltolibrary = new Paths(new string[] { "north" },
                  "north path", "The way to the Library of the University",
                  library);
46              Paths mainhalltoprincipalroom = new Paths(new string[]
                  { "east" }, "east path", "The way to the Principal Room of
                  the University", principalroom);
47              Paths pavementtostreet = new Paths(new string[] { "backward",
                  "south" }, "backward street", "The way to the Elizaberth
                  Street", street);
48              Paths streettopavement = new Paths(new string[] { "forward",
                  "north" }, "forward path", "The way back to the pavement",
                  pavement);
49              Paths librarytomainhall = new Paths(new string[] { "
                  backward", "south" }, "backward path", "The way back to the
                  MainHall", mainhall);
50              Paths principalroomtomainhall = new Paths(new string[]
                  { "backward", "north" }, "Return Main Hall path", "The way
                  back from Principal Room to Main Hall", mainhall);
51              pavement.AddPath(pavementtoMainHall);
52              pavement.AddPath(pavementtostreet);
53              mainhall.AddPath(mainhalltopavement);
54              mainhall.AddPath(mainhalltolibrary);
55              mainhall.AddPath(mainhalltoprincipalroom);
56              library.AddPath(librarytomainhall);
57              street.AddPath(streettopavement);
58              principalroom.AddPath(principalroomtomainhall);
59              //Create some items and put thwm in rooms
60              Item diary = new Item(new string[] { "diary" }, "A principal'
                  s diary", "Thw diary which takes important event.");
61              principalroom.Inventory.Put(diary);
62              Item handledlamp = new Item(new string[] { "handledlamp" }, "
                  A useful handledlamp", "The handledlamp is full of energy
                  and can be used in anytime");
```

```
63              library.Inventory.Put(handledlamp);
64              //Set default location to the player
65              player.Location = pavement;
66              // Set up command handler
67              LookCommand lookCommand = new LookCommand();
68              MoveCommand moveCommand = new MoveCommand();
69              // Print the player name and full description
70              Console.Write($"\nHello, {player.Name}!\n
                  {player.FullDescription}");
71              Console.Write("----------------------------------------");
72              Console.Write($"\n{player.Location.FullDescription}");
73              Console.Write("----------------------------------------");
74              Console.Write(player.Location.PathList);
75              Console.WriteLine("----------------------------------------");
76              //Main loop
77              while (true)
78              {
79                  Console.Write("Enter your commmand: ");
80                  string commandline = Console.ReadLine();
81                  if (string.IsNullOrEmpty(commandline))
82                  {
83                      Console.WriteLine("Please enter the command again.");
84                  }
85                  else if (commandline == "Exit" || commandline == "exit")
86                  {
87                      Console.WriteLine("Thank you for spending time to
                        play SwinAdventure game. See you next time!");
88                      break;
89                  }
90                  else if (commandline.Length > 0)
91                  {
92                      string[] commandsentence = commandline.ToLower().Split
                        ();
93                      if (commandsentence.Length == 3 & commandsentence[0]
                        == "look")
94                      {
95                          string result = lookCommand.Execute(player,
                            commandsentence);
96                          Console.WriteLine(result);
97                      }
98                      else if (commandsentence.Length == 5 &&
                        commandsentence[3] == "in")
99                      {
100                         string result = lookCommand.Execute(player,
                            commandsentence);
101                         Console.WriteLine(result);
102                     }
103                     else if (commandsentence[0] == "help" ||
                        commandsentence[0] == "Help")
```

```
104                     {
105                         Console.WriteLine("Available commands:\n" +
106                               "1. look at <item>\n" +
107                               "2. look at <item> in <container>\n" +
108                               "3. exit - to exit the game\n" +
109                               "4. help - to see this message again");
110                     }
111                     else if (commandsentence[0] == "move" ||
                          commandsentence[0] == "Move" || commandsentence[0]
                          == "Go" || commandsentence[0] == "go")
112                     {
113                         string result = moveCommand.Execute(player,
                              commandsentence);
114                         Console.WriteLine(result);
115                     }
116                     else
117                     {
118                         Console.WriteLine("Unknown command. Please try
                              again.");
119                     }
120                 }
121             }
122         }
123     }
124 }
125
```