```
1 import java.io.File;
2 import java.io.FileNotFoundException;
3 import java.util.Scanner;
4
5 public class Controller
6 {
7    private City[] cityArray = new City[6];
8    private int cityCounter = 0;
9    private Item[] itemArray = new Item[36];
10   private int itemCounter = 0;
11
12   @SuppressWarnings("resource")
13   public Controller(String path)
14   {
15      try
16      {
17         Scanner scFile = new Scanner(new File(path));
18         Scanner sc = null;
19
20         String line;
21         String tID,tDescription;
22         double tItemCost;
23
24         String tName,tDistrict,tHotelName;
25         double tHotelCost;
26
27         while(scFile.hasNext())
28         {
29            int fieldCount = 0;
30            line = scFile.nextLine();
31            sc = new Scanner(line).useDelimiter(",");
32
33            while(sc.hasNext())
34            {
35               fieldCount++;
36               sc.next();
37            }
38
39            sc = new Scanner(line).useDelimiter(",");
40
41            if(fieldCount == 5) //create a new City object
42            {
43               tID = sc.next();
44               tName = sc.next();
45               tDistrict = sc.next();
46               tHotelName = sc.next();
47               tHotelCost = sc.nextDouble();
48               cityArray[cityCounter] = new
City(tID,tName,tDistrict,tHotelName,tHotelCost);
49               cityCounter++;
50            }
51            else //create a new Item object
52            {
53               tID = sc.next();
54               tDescription = sc.next();
55               tItemCost = sc.nextDouble();
56               itemArray[itemCounter] = new
Item(tID,tDescription,tItemCost);
57               itemCounter++;
```

```
58              }
59              //Another approach would be to read in the first 2
fields and check if field 3 is a String (city object must be made))
or a double (item object must be made).
60
61          }
62
63          scFile.close();
64          sc.close();
65
66       }
67       catch (FileNotFoundException e)
68       {
69          System.out.println("File not found.");
70       }
71    }
72
73    public String listAllCities()
74    {
75       String details = "";
76
77       for(int i = 0; i < cityCounter; i++)
78          details += cityArray[i].toString();
79
80       return details;
81    }
82
83    public String listAllItems() //for testing purposes.
84    {
85       String details = "";
86
87       for(int i = 0; i < itemCounter; i++)
88          details += itemArray[i].toString()+"\n";
89
90       return details;
91    }
92
93
94    public String populateList()
95    {
96       String details = "";
97       Item[] tempItemArray;
98
99       for(int i = 0; i < cityCounter; i++)
100      {
101         tempItemArray = new Item[100];
102         int tempCounter = 0;
103
104         for(int j = 0; j < itemCounter; j++)
105         {
106
if(cityArray[i].getStageID().equalsIgnoreCase(itemArray[j].getStageID
()))
107            {
108               tempItemArray[tempCounter] = itemArray[j];
109               tempCounter++;
110            }
111         }
112
```

```
113          if(tempCounter > 0) //Not technically required if we
assume each city will have at least 1 item.
114          {
115             Item[] itemArrayTwo = new Item[tempCounter];
116             for(int z = 0; z < tempCounter; z++) //We could of
course deep copy the array as well.
117                itemArrayTwo[z] = tempItemArray[z];
118
119             cityArray[i].setItemArr(itemArrayTwo);
120          }
121
122          details += cityArray[i].toString();
123       }
124
125       return details;
126    }
127
128
129 }
130
```