```
 1 """
 2 The project should simulate the operation of a
   restaurant.
 3 When I run main.py it should start by randomly
   generating a quantity for
 4 each of the menu items in the restaurant.
 5 These quantities will have to go down when
   customers order the menu items.
 6 Each customer order can contain an Entree, Side,
   Wine, and a Dessert.
 7
 8 Entrees (Random quantity between 1 and 6 of each)
 9 Chicken
10 Beef
11 Vegetarian
12
13 Sides (Random quantity between 5 and 10 of each)
14 Soup
15 Salad
16
17 Wines (Random quantity between 2 and 5 each)
18 Merlot
19 Chardonnay
20 Pinot Noir
21 Rose
22
23 Desserts (Random quantity between 1 and 3 each)
24 Flan
25 Creme Brulee
26 Chocolate Moose
27 Cheesecake
28
29 After generating the random quantities of each menu
    item
30 the program should prompt the user to input a role
   and an action
31 (this does not need to be done as a single input).
32
33 The roles and the actions each role can take are
   defined below
34
```

```
35 Waiter
36 Read Menu: lists how much of each menu item is
   available
37 What are the (Entrees/Wines/Sides/Desserts): lists
   how much of each of the chosen categories is
   available
38
39 Customer
40 Order (ordered food): takes an order from the
   customer and subtracts the order from the available
    food total.
41 Customers are told if what they ordered is not
   available and asked if they would like something
   else
42 Customers must be able to make their selection in
   any order and leave out choices. For example, they
   could just ask
43 for Merlot or they could ask for Flan, Pinot Noir,
   Salad, and Beef
44 Random Choice
45 A random order is made using the choice function
   from the random module
46 See Chapter 11 of IP or 3.11 of the Python Cookbook
47
48 Manager
49 Close: Lists the remaining food at the end of the
   night and then sets all of the values to zero
50 Open: Restarts the main .py file and generates a
   new random amount of foods
51
52 """
53 from MenuFull import MenuFull
54 import time
55
56 # generate our menu
57 # it does all the stuff for us in the class init!
58 menu = MenuFull()
59
60 # start our main game loop
61 cafeopen = True
62 while cafeopen:
```

```
63      role_select = input('Welcome to the Hades Cafe
    . Choose a role by typing\n'
64                          'WAITER, CUSTOMER, MANAGER
    or QUIT: \n'
65                           '>>>  ')
66      if role_select.casefold() == 'quit':
67          print("You escaped the sisyphean cycle. Or
     have you?")
68          cafeopen = False
69          break
70      elif role_select.casefold() == 'waiter':
71          print("You're the waiter!")
72          while iswaiter := True:
73              waiterprompt = input('Options(type one
    ): MENU, ENTREES, SIDES, WINES, DESSERTS, or ROLE\
    n'
74                                   '>>>  ')
75              if waiterprompt.casefold() == 'role':
76                  iswaiter = False
77                  break
78              elif waiterprompt.casefold() == 'menu'
    :
79                  menu.read_menu()
80              elif waiterprompt.casefold() == '
    entrees':
81                  menu.Entrees.list_category()
82              elif waiterprompt.casefold() == 'sides
    ':
83                  menu.Sides.list_category()
84              elif waiterprompt.casefold() == 'wines
    ':
85                  menu.Wines.list_category()
86              elif waiterprompt.casefold() == '
    desserts':
87                  menu.Desserts.list_category()
88              else:
89                  print("Let's try that again...")
90
91
92
93
```

```
 94     elif role_select.casefold() == 'customer':
 95         print("You're the customer!")
 96         while iscustomer := True:
 97             custprompt = input('Type your order
    choices, each separated by a single "," and no
    spaces.\n'
 98                                "Please note any
     '-' hyphens in menu names.\n"
 99                                'Type RANDOM for a
    meal selected by chance.\n '
100                                '>>>  ')
101             orderlist = custprompt.split(',') #bc
    can type in any order, feed em into a list
102             # iterate over that list to pull each
    out one by one for comparisons
103             for x in orderlist:
104                 x.casefold()
105                 if x=='chicken' or x=='beef' or x
    =='vegetarian':
106                     menu.Entrees.check_order(x)
107                 elif x=='soup' or x=='salad':
108                     menu.Sides.check_order(x)
109                 elif x=='pinot-noir' or x==
    chardonnay' or x=='merlot' or x=='rose':
110                     menu.Wines.check_order(x)
111                 elif x=='flan' or x=='creme-brulee
    ' or x=='chocolate-mouse' or x=='cheesecake':
112                     menu.Desserts.check_order(x)
113                 elif x == 'random':  # whenever we
     see random, generate a whole meal
114                     menu.Entrees.random_pick()
115                     menu.Sides.random_pick()
116                     menu.Wines.random_pick()
117                     menu.Desserts.random_pick()
118                 else:
119                     print("We don't serve {} here
    .".format(x))
120
121             custprompt2 = input('\n'
122                                 'Type anything you
     want to order MORE\n'
```

```
123                                     'or type ROLE to
    return to selection \n'
124                                 '>>>  ')
125             if custprompt2.casefold() == 'role':
126                 ismanager = False
127                 break
128 
129     #  Here's where we can pretend to be a manager
    !
130     elif role_select.casefold() == 'manager':
131         print("You're the manager...")
132         while ismanager := True:
133             mgrprompt = input('Options(type one):
    CLOSE, OPEN, or ROLE\n'
134                                       '>>>  ')
135             # return to role selection
136             if mgrprompt.casefold() == 'role':
137                 ismanager = False
138                 break
139             # close the store as per project
    outline
140             elif mgrprompt.casefold() == 'close':
141                 menu.read_menu()
142                 menu.close()
143                 print("All menu items have been
    destroyed.")
144             # open the store as per project
    outline
145             elif mgrprompt.casefold() == 'open':
146                 ismanager = False
147                 menu.reset()
148                 print("Regenerating menu and
    reopening.")
149                 time.sleep(1)
150                 print("..")
151                 time.sleep(1)
152                 print("..")
153                 time.sleep(1)
154                 print("..")
155                 break
156 
```

```
157
158     # catch those misspellings and re-prompt at
    top of loop
159     else:
160         print("Let's try again...\n")
```