Food Magnate Simulation

MARKS /50

Programming Theory Questions

These questions refer to the preliminary material and require you to load the skeleton program, but <u>do not</u> require any additional programming.

1. State the name of an identifier for:
   a) An attribute in the `Household` class that would **not** be instantiated for each new object [1]
   b) A subroutine in the `Settlement` class that accepts parameters by reference [1]
   c) A subclass [1]
   d) A local variable that is used to return a Boolean [1]
   e) Two subroutines from the `Company` class that **cannot** be called from outside the `Company` class [2]
   f) A library string function called from the `GetIndexOfCompany` subroutine in the `Simulation` class [1]
   g) A collection attribute in the `Company` class [1]
   h) An instance of `Settlement` [1]

2. Showing and explaining your working, give the probability of a call to `ProcessCostOfFuelChangeEvent` being made from the `DisplayEventsAtDayEnd` subroutine in the `Simulation` class. [3]

3. Explain how validation might be added to the `OpenOutlet` subroutine of the `Company` class to prevent a new outlet being created beyond the bounds of the settlement. You do **not** need to write any code. [3]

4. Each `Household` object is stored within an ArrayList called `Households`. Describe how a *Dictionary* could have been used instead to store `Household` objects. [3]

5. Describe in full how the `GetDistanceBetweenTwoOutlets` subroutine of the `Company` class calculates the distance between two outlets. [4]

6. Explain the role of the object of type `Random` in the `Household` class. [2]

7. Explain the role of the variable `UpOrDown` in the `ProcessCostOfFuelChangeEvent` subroutine of the `Simulation` class. [3]

8. In the `Simulation` constructor, the integer literals 100000, 200 and 203 are passed to the `Company` constructor when creating the 'AQA Burgers' company. State the role of each of these integer literals. [3]

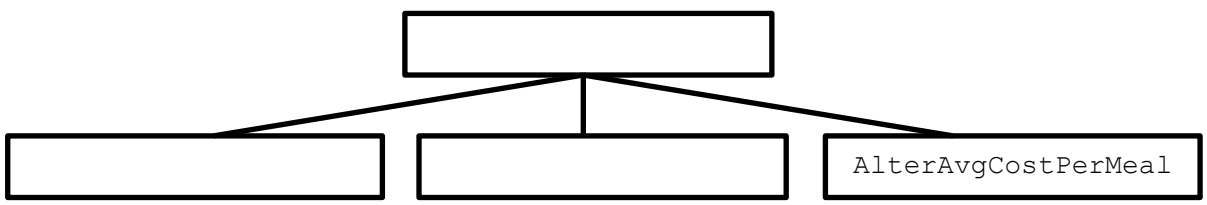
9. Describe in full the operation of the `GetIndexOfCompany` subroutine in the `Simulation` class. [5]

10. Describe the circumstances under which the `ModifyCompany` subroutine of the `Simulation` class would output the text 'Invalid coordinates'. [3]

11. Currently, a call to the `LargeSettlement` constructor could not result in a settlement that is smaller than 1,000 by 1,000. This is true even if negative numbers are entered by the user when prompted for additional x and y values. Explain how a call to the `LargeSettlement` constructor never results in a smaller settlement size. [3]

12. Describe the concept of constructor overloading, and explain how constructor overloading could have been used instead of inheritance for the creation of a new large settlement. [4]

13. Complete the following hierarchy chart for part of the `Simulation` class of the Skeleton Program. You should **not** include calls to any library subroutines. [3]



14. Describe how the program would respond to a call to the `Company` constructor using a category that is neither 'fast food', 'family' nor 'named chef'. [2]