

## Exercise Set 2: Simple Classes

**Exercise 2.1** A class to represent information about one person.

- Define the member data for the **Class Person** that will include the date of birth (year only). Do not use more than seven attributes.
- Define the constructor and the `toString` method for the **Class Person**.
- Develop the purpose and the contract (header) for at least three methods for the **Class Person**.
- Develop the method `isOlderThan` that determines whether this person is older than a given age.
- Draw the UML diagram for this class.

**Exercise 2.2** A class that represents weather data for one day.

- Run the code for the tests of the **Class WeatherData**.
- Add the code to test the method `tempPhrase`.
- Develop the method `howWet` for the **Class WeatherData**. It should return the String `"dry"` if the precipitation is below 0.01, `"wet"`, if the precipitation is below 0.25, `"soggy"` if the precipitation is below 1.0 and `"very wet"` otherwise.

**Exercise 2.3** A class that represents automobiles.

- Design the **Class Car** that records the make and model of the car, the fuel tank capacity in gallons, and the estimated fuel consumption given in miles per gallon.
- Develop the constructor and the `toString` method for this class.
- Develop the method `maxDistance`, which computes the distance the car can travel on one tank of gas.
- Develop the method `canReach`, which determines whether a destination (distance given in miles) is reachable on one tank of gas.
- Develop the method `goFartherThan`, which determines whether this car can travel farther on one tank of gas than some other given car.
- Draw the UML diagram for this class.

**Exercise 2.4** A class that represents one type of item in a grocery store (a can of coffee).

- Design the **Class Coffee** that represents a coffee can in the grocery store. The relevant information is the brand name, the weight of the can, given in ounces, and the price of the can, given in cents.

- Develop the constructor and the `toString` method for this class.
- Develop the method `unitPrice`, which computes the price per ounce of this grocery item.
- Develop the method `isCheaperThan`, which determines whether the unit price is lower than some given price.
- Develop the method `betterPriceThan` for this class, that determines whether this coffee is cheaper (in terms of the unit price) than some other given `Coffee`.
- Draw the UML diagram for this class.

**Exercise 2.5** Run the existing code for the tests of the Class `DayData`. Complete the test suite for the Class `DayData`.

**Exercise 2.6** A class that represents inventory information about an item in a grocery store (a can of coffee).

- Design the Class `InventoryItem` that keeps a record of the number of specific `Coffee` items in stock, as well as the number that has been sold.
- Develop the constructor and the `toString` method for this class.
- Develop the method `itemValue`, which computes the value of the current stock of `Coffee` items.
- Develop the method `grossIncome`, which determines the amount of money received for the `Coffee` items already sold.
- Draw the UML diagram for this class.

**Exercise 2.7** Continuation of Exercise 2.1.

- Define the Class `Address`, which contains the information about the city, state, and zip code.
- Modify the Class `Person`, so that it also contains person's address. Remember to build the constructor and the `toString()` method, and test them.
- Design the method `getState()` in the Class `Address`.
- Design the method `getState()` in the Class `Person`. Remember to follow the *Law of Demeter*.