10 EC major assessment points

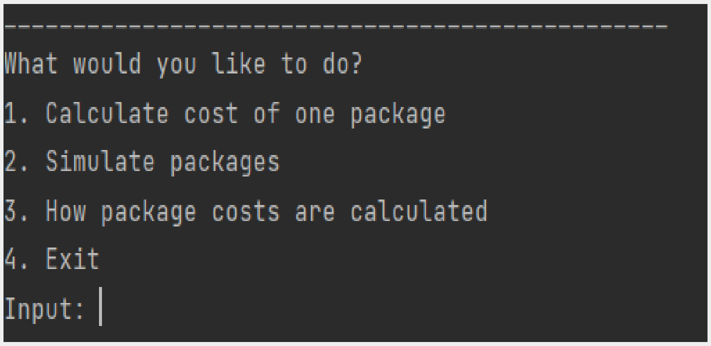
Package Shipping Simulator Overview

You are being tasked with writing an application that will help calculate shipping costs for various packages and simulate the total cost for a variable number of packages that may be shipped during a busy time of the season.

Shipping costs are calculated using the source and destination zip code, and the weight of the package.

Your application should support the ability to simulate a large number of packages, and to allow the user to enter package information to determine the cost of shipping.

When your program starts, the user should be presented with 3 options:



```
------------------------------------------------
What would you like to do?
1. Calculate cost of one package
2. Simulate packages
3. How package costs are calculated
4. Exit
Input: |
```

Option 1: Calculate the cost of one package. Here is a picture of how this option should function.

Option 2: Simulate some number of packages. Here is a picture of how this option should function.

Option 3: An informational menu item that will show the user exactly how packages are being calculated in an easy to read format.

Detailed information about how each package is simulated and calculated are below in the **Technical Specifications** section!

Technical Specifications on next page

Technical Specifications (Address class)

The `Address` class should have the following information:

- The house number (remember that this could be a number like 101 or it could also have a letter like 101A)
- The name of the street
- The apartment number (which is optional)
- The city
- The state
- The zip code (which should be exactly 5 numbers)

The `Address` class should have three constructors:

- One that takes in the separate pieces of the address and assigns them to the proper fields
- One that takes in an `Address` object and copies its information into the fields
- One that takes in an `Address` as a full `String`, breaks up the information, and copies into into the fields

The `Address` class method specification:

- Accessor and mutator method for each field
- `toString()` method that returns the address in the following format:
  - `Number Street Apt Number (optional), City, State Zip`
  - *Example*: `1313 Mockingbird Lane Apt 6A, Springfield, MO 65123`

Technical Specifications (Package class)

The `Package` class should have the following information:

- Two `Address` objects (representing an origin and a destination address
- The package should also should have a weight in pounds, a length, height, and width in inches (can be decimals)
    - A package cannot be less than a tenth of a pound or smaller than 2 inches on each side
- The `Package` class should have accessor methods for all of its fields

Technical Specifications (PostageCalculator class)

The `PostageCalculator` class is designed to be able to calculate how much a package will cost to ship

- The formula to determine the cost of shipping is:
  - The base cost is $3.75
  - Add 5 cents for each tenth of a pound
  - Divide the difference in county codes by 100 and add that to calculate the final total
    - The county code is the first 3 digits f the zip code

- Accounting for oversize packages:
  - Any package that exceeds a combined 36 inches will cost an additional 10 cents per inch.
  - Additionally, packages that are more than 40 pounds will be charged 10 cents per tenth of a pound over the 40 pound limit.

- The `PostageCalculator` class is a **static library** with 3 different versions of the `calculatePostage` method:
  - Each version should return the cost of shipping
  - One version of `calculatePostage` should take in the 2 zip codes, weight of the package, and the dimensions of the package
  - One version of `calculatePostage` should take in the 2 `Address` objects, the weight of the package, and the dimensions of the package.
  - One version of `calculatePostage` should simply take in a `Package` object.
  - You may add other methods here as you see fit.

Technical Specifications (PackageSimulator class)

The `PackageSimulator` class is designed to be allow the user to simulate some number of packages

- A `PackageSimulator` object should contain an `ArrayList` of `Package` objects that starts off as empty.
- It should contain a method called `generatePackages` that takes in an integer as a parameter that represents how many packages to generate.
  - The package generated should be "smart." You need to generate real zip codes and create packages based on this information.
  - For this simulation, make the addresses domestic (US only)
  - You will need to research zip codes and general averages for package sizes and weights.
- It should contain a method called `generateTotalCost` that calculates the cost of every package simulated and returns it as a double.
- It should contain a method called `getSimulationInfo` that returns all of the simulation info in the format specified at the beginning of the document.
- It should contain a method called `resetSimulation` that will reset the ArrayList of Packages to empty.

Finally, create a main menu according to the specifications at the top of the document in a Runner class.