CS146: Quiz 10
Due Tuesday, April 18, at 7:00AM
10 points

You will solve the fractional knapsack problem using the greedy algorithm discussed in class. **To get started, import the starter file, Knapsack.java into the greedy package you create in a new Java Project.** Please do not change any of the method signatures in the class. Implement the methods described below. You are free to test your code however you prefer. These methods should be completed individually using any IDE you are comfortable with. You are free to use the textbook, slides, class notes, and the Java API Documentation, but **DO NOT** consult any other resources.

Solving the Fractional Knapsack Problem

The fractional knapsack problem as follows. A thief robbing a store wants to take the most valuable load that can be carried in a knapsack capable of carrying at most W pounds of loot. The thief can choose to take any subset (including fractional amounts) of n items in the store. The w_i item is worth v_i dollars and weighs w_i pounds, where v_i and w_i are integers. Which and how much of each item should the thief take?

The idea is to take the item that costs the most per pound first and then proceed on to the item that costs the second most per pound, and so on until you have reached the capacity that you can carry in the knapsack. Remember you can take a fractional part of an item.

```
public static ArrayList<Double> fractionalKnapsack(ArrayList<Item> items, double maxWeight)
```

This method should return how much of each item (as a fraction) from the ArrayList of Items that you should take. The entries in the ArrayList should be values between 0.0 and 1.0.

- If I take **all of an item**, the corresponding entry in the returned ArrayList should be **1.0**
- If I take **none of an item**, the corresponding entry in the returned ArrayList should be **0.0**.

You can assume that the value of items is already ordered by cost per pound with the largest cost per pound in the first entry of the items. This assumption is for the inputs I will test your code on. When you are testing your code, you will need to make sure that items are ordered by cost per pound with the largest item in the first entry of the ArrayList.

Submission

Please create a jar or zip file of your project. It should include the java files (Knapsack.java) and submit it on Canvas. If your file includes class files or other extraneous files outside of the package folder and java files, you will receive an automatic one point deduction.