

CSCI 3005 – Spring 2020 – Programming Assignment 1

Your assignment consists of implementing a dynamic programming solution to the 0-1 knapsack problem. Your solution must be stored in the *DPKnapsack* class. Your class is required to have the following methods:

- *DPKnapsack(int capacity, String itemFile)*: a constructor to accept the default capacity and the name of a data file as arguments. The data file will have several lines of the form below (the first entry in each line is a string, the other two are integer values):

```
itemName itemWeight itemValue
```
- *int optimalWeight()*: returns the overall weight of the items in the optimal solution subset
- *int optimalNumber()*: returns the number of items in the optimal solution subset
- *boolean contains(String item)*: returns true if *item* is in the optimal solution subset, false otherwise
- *String solution()*: returns a string representing the optimal solution subset. The precise format of the string is left to you. Output which is appropriately labeled and formatted will obviously receive a better grade. Obviously, the solution string must contain each item in the knapsack exactly once.

In addition, the methods below, which overload the ones already presented, require that your program solve the problem using the same items, but with a different knapsack capacity.

- *int optimalWeight(int maxWeight)*: returns the overall weight of the items in the optimal solution subset for a knapsack with *maxWeight*.
- *int optimalNumber(int maxWeight)*: returns the number of items in the optimal solution subset for a knapsack with *maxWeight*.
- *boolean contains(String item, int maxWeight)*: returns true if *item* is in the optimal solution subset for a knapsack with *maxWeight*, false otherwise.
- *String solution(int maxWeight)*: returns a string representing the optimal solution subset for a knapsack with *maxWeight*.

Naturally, you may develop additional private methods in the process of implementing your solution, but the methods above are required. Your solution should be stored in the **DPKnapsack.java** source file. Comments for your class and its methods should follow *javadoc* guidelines. The source file **DPKTest.java** is provided to test some of the functionality of your class. However, you probably want to design additional testing cases before submitting your program for grading.

When done, submit your DPKnapsack.java solution (as well as any other .java files you write) to Mimir for testing. Your submission should contain neither DPKTest.java nor any .txt files.