

Assignment Description: Dungeon Looter, Dynamic Programming Application (75 points) The premise of the DungeonLooter program is to provide a game mechanic that will showcase the use of Optimal Change Making and [0-1] Knapsack. In this “game” an *Adventurer* is given a finite carrying weight. They co-exist in a world with various treasure *Chests* that contain different types of *Items*, each with its own combination of *weight and value*. The objective is for the adventurer to collect the most valuable items from each chest as they can, so they can sell their inventory in exchange for currency. This is an implementation of the [0-1] knapsack problem introduced in class. Given a set of items (the chest’s contents) and a knapsack capacity (the adventurers current remaining carrying weight), identify the items that result in the most value (remove those items from the chest and add them to the adventurer’s inventory).

Next, after the adventurer has performed their looting of chests, they can *sell* their inventory. This means to remove those items from their inventory and be given the appropriate amount in the proper denomination. Because this is a fantasy world there are oddly valued denominations such as The 13 valued Silver or the 32 valued Diamond. This is an implementation of the optimal change making algorithm shown in class. Given a target value (the total value of the inventory being sold) identify the set of denominations to optimally satisfy that amount (add those denominations to the adventurer’s coin purse.)

Starter Code has been provided with some basic methods implemented for the students.

Any methods, classes, fields and functions can be implemented as desired by the student, however the sample code in lines 240-266 of the provided file must run (this is the required interface).

In summary those unique calls are:

```
player = Adventurer(carry_weight=100)
game = Game(player)
game.show_player_inventory()
game.add_chest(Chest())
game.show_chests()
game.loot_chests()
game.sell_items()
```

Submission Instructions Please submit a single python file (.py) with a class defined as described blow. Please name the file:

DungeonLooter.py

The file is commented with explanations of each class and method. If there are any questions about the meaning of the provided code or implementation details, please let me know.