|  |  |  |
| --- | --- | --- |
| ***Computer Science 2061 -*  ASSIGNMENT #09** | | |
| **Program** | Description | **Execution Listing** |
|  | Write a program that defines a dictionary *wallet* that initially contains the following key/value pairs:   |  |  | | --- | --- | | **Key** | **Value** | | money | A dictionary that contains key value pairs to store information about the quantity of bills in the wallet, including:   * two $20 * one $10 * zero $5 * Four $1 | | creditCards | A list that initially contains the following credit cards:   * Visa * Discovery * MasterCard | | IDs | A list that initially contains the following IDs:   * Drivers License * Student ID |   Display the contents of the dictionary as shown to the right.  Now, make the following changes to the dictionary:   * Add a new key/value pair to wallet with these values:  |  |  | | --- | --- | | **Key** | **Value** | | coupons | A dictionary that contains key value pairs to store information about the coupons including:   * Cub, meat * Kohls, shirt * Target, toothpaste |  * Add 3 $20 bills to the wallet * Remove the ‘Discovery’ credit card   Now display the contents of the wallet again. | *Wallet Contents:*  *Credit Cards:*  *Visa*  *Discovery*  *MasterCard*  *IDs:*  *Drivers License*  *Student ID*  *Total cash is: $54*  *After changes, wallet contents are:*  *Credit Cards:*  *Visa*  *MasterCard*  *IDs:*  *Drivers License*  *Student ID*  *Coupons:*  *Target toothpaste*  *Kohls shirt*  *Cub meat*  *Total cash is: $114* |

|  |  |  |
| --- | --- | --- |
|  | Write a program that initially defines the following dictionaries. In each dictionary, make the name of the item (e.g. ‘potato’) the key  **shoppingList**– contains items in a shopping list including   * 2 potatoes * 1 onion * 5 lettuce.   **inventory** – contains the items in a grocer store inventory including:   * 6 potatoes * 0 lettuce * 32 onions * 15 carrots   **prices** – contains the prices of the items in the inventory:   * potato - $4/each * lettuce - $2/each * onion - $1.5/each * carrot - $3/each   Write a function *printInventory()* that will display the contents and prices of the inventory as shown to the right. Pass the necessary parameters.  Write a function called *computeBill()* that will calculate the shopping bill as shown to the right. Pass the necessary parameters. As shown, this function will take the items from the shopping list and extract the necessary information from inventory and prices to display the results. If an item is out of stock, it should show it.  **NO GLOBAL VARIABLES! NO NESTED FUNCTIONS!** You need to pass parameters**.** | *\*\*\*\*\*\*\*\*\*CUB Foods Inventory\*\*\*\*\*\*\*\*\*\**  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  *Item Price Quantity Value*  *Carrot $3 15 $45*  *lettuce $2 0 $0*  *onion $1.5 32 $48.0*  *potato $4 6 $24*  *Your shopping bill is:*  *lettuce - out of stock.*  *1 onion at $ 1.5 each - total $1.5*  *2 potato at $ 4 each - total $8*  *The total bill is $ 9.5* |

|  |
| --- |
| **Notes** |
| 1. In the execution listing, numeric values don’t have to appear exactly as shown above. For example, $44.50 is OK instead of $44.5. 2. Make sure that you use good programming style and documentation. 3. **USE VARIABLES FOR ALL VALUES IN THE PROGRAM**. 4. Unless instructed otherwise, always use the test data described in the problem. 5. Copy and paste your execution and code listings into the *Homework Template* document. Please ensure that:    * Problems are in the order listed on the assignment sheet (I.E. Program #1 first, Program #2 second, etc.)    * The execution listing corresponds with the code listing. In other words, the code listing will produce the execution listing. If it does not, no credit will be received for that problem. 6. The work you submit must be your own. Please note that any of the following actions are considered cheating:    * Electronically copying or inserting any code that you did not create.    * Manually copying or inserting any code that you did not create. |