

Main:

- Reads in .csv file and stores it into a BST
- Program then asks for user input (budget)
- Asks to either edit database or build a computer
 - Edit: change the information in the database by many different specifications
 - Build a computer:
 - 4 options:
 - Add parts to shopping cart
 - Asks to enter name of part given the list of parts
 - Object is added to a BST called shopping cart
 - Cost is removed from budget
 - Remove parts from shopping cart
 - Asks to enter name of item of shopping cart to be deleted
 - Object is removed from shopping cart and cost is added back to budget
 - View shopping cart
 - Change the budget

readFromFile():

- Creates ifstream inFile and reads .csv file
- Opens .csv and checks if the file is found or not
- While loop
 - Creates new CompPart item, reads data into item as a CompPart item, adds CompPart item into BST and HashTable

editDatabase:

-Add new Part

- Creates a new CompPartWrapper
- User chooses which part they want to add
 - Asks the user to enter data
 - name->price->manufacturer->powerConsumption->performanceIndex->Compatibility
 - Prints out the item that user added
 - Adds the part to the BST and HashTable

-Remove Part

- user enters the part name they want to delete
- program calls the HashTable delete function and the BST delete function
- throws error if the part is not part of the database

- Search

- user inputs string
- Program looks through hashtable for the string.
- Throws error if cannot find part

-listHashSequence:

- traverses the hashtable and prints in that order

-listKeySequence:

- Traverses the hash table using the index and printing them out in that order

- listPrice:

- switch statement to print by price or price indented
- bsthandler calls displayListByPrice or displayListByPriceIndented, depending on what the user chose

-listPerformance:

- switch statement to print by performance or performance Indented
- Bsthandler calls displayListByPerformance or displayListByPerformanceIndented

-PrintsEfficiency:

- switch statement to choose either PriceBST, PerformanceBST, or HashTable efficiency

add PartToCart():

- Choose with computer part you would like to add
 - Can sort by price/performance
 - switch statement for choice
 - singlyLinkedList is created to show the available parts
 - Takes user input, gets the item and puts it into shopping cart
 - CompPartWrapper part created and the item associated with the input is assigned to part
 - shoppingCart adds part
 - Budget gets subtracted with the cost of the part
 - Outputs user's selection

removePartsFromCart():

- Chooses which computer part you would like to remove from shopping cart
 - Prints out current shopping cart using inorder traversal of BST
 - Types in the name of the item
 - CompPartWrapper part created and the item associated with the input is assigned to part
 - shoppingCart adds part
 - Budget gets subtracted with the cost of the part
 - Outputs user's selection

viewCart():

- Prints the shopping cart BST using inorder traversal
- Returns void

changeBudget():

- Asks user to enter a new budget
- Inputs the value as the budget
- Returns void

saveCart():

- Returns void
- Saves the shopping cart into a file
- Uses BSTHandler to open file
 - open file, close, open app, print heading bst file, write shopping cart in order
traverse to file, close file