Practice Problem | Shopping Cart

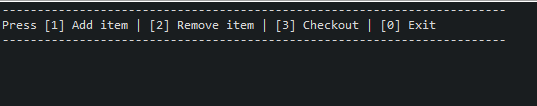
# Problem

Create a store application that allows a customer to perform the following:

* Add items to cart
* Remove items from cart
* Checkout

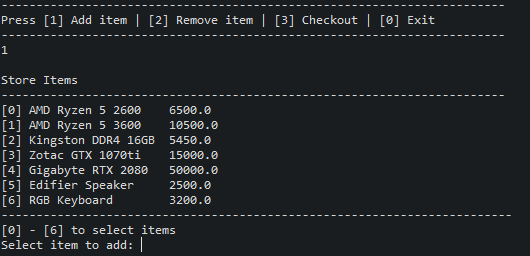
## Prompt Screen

Start of application will show a prompt screen to ask the customer which action to be performed.



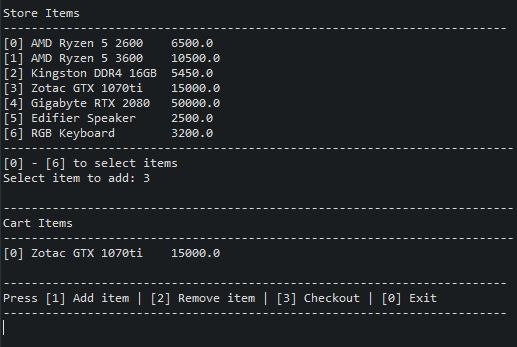
## Add item

The customer is shown a list of available items to select. These store items areread from a file called **store-items.csv**. A customer can add the same item as many times as they want.



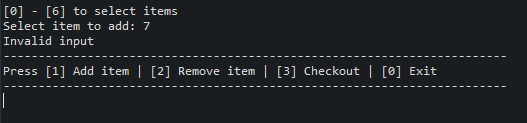
### Normal Condition

Selecting an available item will show the cart items and return back to the **Prompt Screen**.



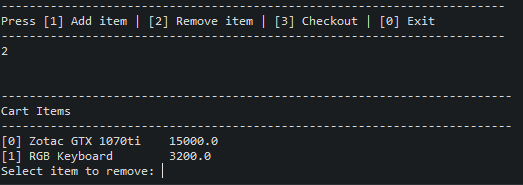
### Invalid Input

When an input entered that is not within the specified value, an **“Invalid input”** message is displayed and returns back to **Prompt Screen**



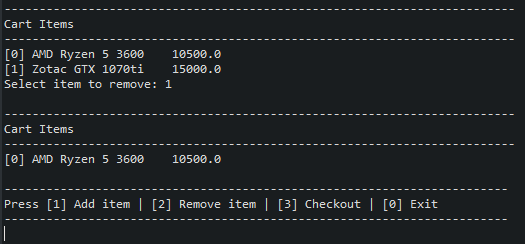
## Remove Item

Customer will select the item they want to remove by the corresponding number associated with the item



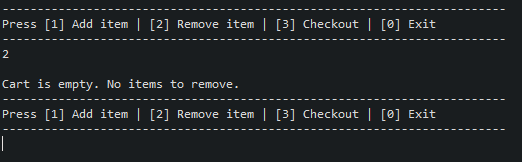
### Normal Condition

When a valid input is entered, the updated **Cart Items** are displayed and returns to **Prompt Screen**

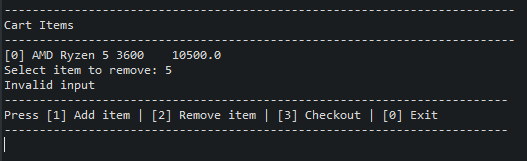


### Empty Cart

When there’s no item in the cart, **“Cart is empty. No items to remove.”** Is displayed.



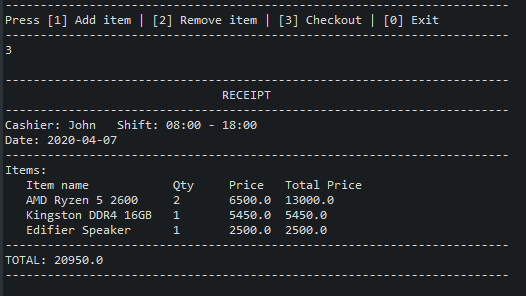
### Invalid Input



## Checkout

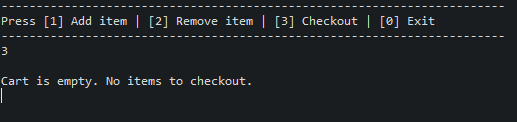
### Normal

When the customer checkout, the receipt is displayed and a file **receipt.txt** is created. The program terminates after checkout. The receipt contains the cashier’s name and shift, the date today, list of items, and the total price.



### Empty Cart

When there are no items in the cart, **“Cart is empty. No items to checkout.”** isdisplayed.



## File Content Sample

### store-items.csv

AMD Ryzen 5 2600,6500.00

AMD Ryzen 5 3600,10500.00

Kingston DDR4 16GB,5450.00

Zotac GTX 1070ti,15000.00

Gigabyte RTX 2080,50000.00

Edifier Speaker,2500.00

RGB Keyboard,3200.00

### receipt.txt

------------------------------------------------------------------------

RECEIPT

------------------------------------------------------------------------

Cashier: John Shift: 08:00 - 18:00

Date: 2020-04-07

------------------------------------------------------------------------

Items:

Item name Qty Price Total Price

AMD Ryzen 5 2600 2 6500.0 13000.0

Kingston DDR4 16GB 1 5450.0 5450.0

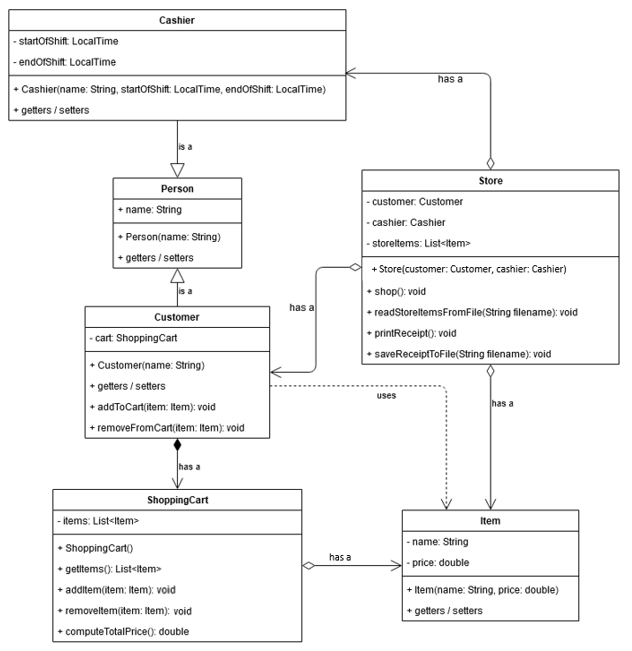
Edifier Speaker 1 2500.0 2500.0

------------------------------------------------------------------------

TOTAL: 20950.0

------------------------------------------------------------------------

## UML Class Diagram



## Class Definition

|  |  |  |
| --- | --- | --- |
| Class | Method | Description |
| Store | shop() | The main flow of the program is defined in this method. It must have the scanner to get inputs from user |
| readStoreItemsFromFile(filename) | This method retrieves the store items data from **store-items.csv** |
| printReceipt() | Prints the receipt with System.out.print() on the screen |
| saveReceiptToFile(filename) | Saves the receipt to **receipt.txt** file |

The main method should contain a customer, a cashier object, and a store object.

The customer and cashier object will be passed as an argument to the store object’s constructor.

The **Store** class **shop()** should be the only method called in the main.