Fruit Juice Machine

Prerequisite : C++ Compiler/ IDE should be setup successfully on the system

Problem Statement

- A new fruit juice machine has been purchased for the cafeteria, and a program is needed to make the machine function properly. The machine dispenses apple juice, orange juice, mango lassi, and fruit punch in recyclable containers. In this programming example, we write a program for the fruit juice machine so that it can be put into operation.
- The program should do the following:
  1. Show the customer the different products sold by the juice machine.
  2. Let the customer make the selection.
  3. Show the customer the cost of the item selected.
  4. Accept money from the customer.
  5. Release the item.

Analysis of the problem

- Define problem's main objective
- Define objects' properties
- Define objects' functionality

Objects (Focus on nouns)

- A new fruit juice machine has been purchased for the cafeteria, and a program is needed to make the machine function properly. The machine dispenses apple juice, orange juice, mango lassi, and fruit punch in recyclable containers. In this programming example, we write a program for the fruit juice machine so that it can be put into operation.
- The program should do the following:
  1. Show the customer the different products sold by the juice machine.
  2. Let the customer make the selection.
  3. Show the customer the cost of the item selected.
  4. Accept **money** from the customer.
  5. Release the item.

Objects' Properties And Operations (Focus On Verb)

- A new fruit juice machine has been purchased for the cafeteria, and a program is needed to make the machine function properly. The machine dispenses apple juice, orange juice, mango lassi, and fruit punch in recyclable containers. In this programming example, we write a program for the fruit juice machine so that it can be put into operation.
- The program should do the following:
  1. Show the customer the different products sold by the juice machine.
  2. Let the customer make the selection.
  3. Show the customer the cost of the item selected.
  4. Accept money from the customer.
  5. Release the item.

Cashier/ Locker Class

- Member Variables :
  - Cash in the registry
- Member functions:
  - Increment the amount when customer deposit the money : depositAmount
  - Get the current balance of the registry: getCurrentBalance
- Registery should have initial balance.

Product class

- Member variable:
  - Cost of the product
  - Total count of the product items
- Memeber functions:
  - Get the product cost : getCost
  - Get the total count of the product : getItemCount
  - Decrement the count of the product when customer purchase the product : releaseItem
- Initialize the product with initial item count and cost

Main Class

Algorithm :

Step 1 : Check that our inventory is not empty

Step 2 : Show the list of products to the customer

Step 3 : Ask cusomter to choose the product

Step 4 : Show the customer the cost of the product

Step 5 : Ask customer to deposit the amount

Step 6 : If the deposited amount is less than cost of the product

Step 7 : Ask customer to deposit the remaining amount till the amount is not greater or equal to the cost of the product.

Step 6.a : If the deposited amount is greater than cost of the product

Step 8 : Propmt the message to collect the change.

Step 6.b : if the deposited amount is at least to the cost of product :

Step 9 : Add the amount to the registery

Step 10 : release the product by decrementing that product count and display the appropriate message.

Step 11 : Go to Step 1