# Laboratory 01 Designs

IceCreamContainer

- containerType : char

- containerMax : double

- cupPrice : double

- conePrice : double

- waffleConePrice : double

- iceCreamRate : int

- hotToppingRate : int

- candyRate : int

- cupMax : double

- coneMax : double

- waffleConeMax : double

- pricePerHectoGram : double

- iceCreamMass : double

- hotToppingMass : double

- candyMass : double

- totalMass : double

+ IceCreamContainer(filename : String) : void

Loads the prices, dispensing rates, and maximum capacities from the price file.

+ AddIceCream(numSeconds : double) : void

Adds the correct amount of ice cream accounting for the maximum mass.

+ AddHotTopping(numSeconds : double) : void

Adds the correct amount of hot topping accounting for the maximum mass.

+ AddCandy(numSeconds : double) : void

Adds the correct amount of candy accounting for the maximum mass.

+ SelectContainer(containerChoice : char) : void

Sets the container type as long as there is no mass already.

+ GetMass(void) : double {totalMass}

Returns the mass of materials in the container.

+ GetIceCreamMass(void) : double {iceCreamMass}

Returns the mass of ice cream dispensed.

+ GetHotToppingMass(void) : double {hotToppingMass}

Returns the mass of hot topping dispensed.

+ GetCandyMass(void) : double {candyMass}

Returns the mass of candy dispensed.

+ ComputeSubtotal (void) : double {subTotal}

Computes and returns the sub total based up container type and the mass/pricing.

SellIceCream

None.

+ main(args : String[ ]) : void

Fills the container, computes subtotal, computes tax, computes total, and displays receipt.

+ FillTheContainer(aContainer : IceCreamContainer) : void

Allows the user to select container and add items to that container until they quit.

+ GetChoice(void) : char {choice}

Displays a menu of options and gets/returns the user's choice.

+ PerformChoice(choice : char, aContainer : IceCreamContainer) : void

Carries out the user's choice.

+ GetContainerChoice(void) : char {containerChoice}

Gets and returns the user's choice of container.

+ DispenseIceCream (aContainer : IceCreamContainer) : void

Gets the number of seconds of dispensing and adds that amount of ice cream.

+ DispenseCandy (aContainer : IceCreamContainer) : void

Gets the number of seconds of dispensing and adds that amount of candy.

+ DispenseHotTopping (aContainer : IceCreamContainer) : void

Gets the number of seconds of dispensing and adds that amount of hot topping.

+ GetSeconds(void) : double {seconds}

Gets and returns the number of seconds of material dispensed.

+ ComputeTax (subTotal : double) : double {tax}

Computes and returns the tax based upon the subtotal.

+ ComputeTotal (tax : double, subTotal : double) : double {total}

Computes and returns the total due.

+ DisplayReceipt (aContainer : IceCreamContainer, tax : double, subTotal : double, total : double) : void

This displays all elements with masses and costs as well as a subtotal, tax, and total.