# COMP10062: Assignment 2

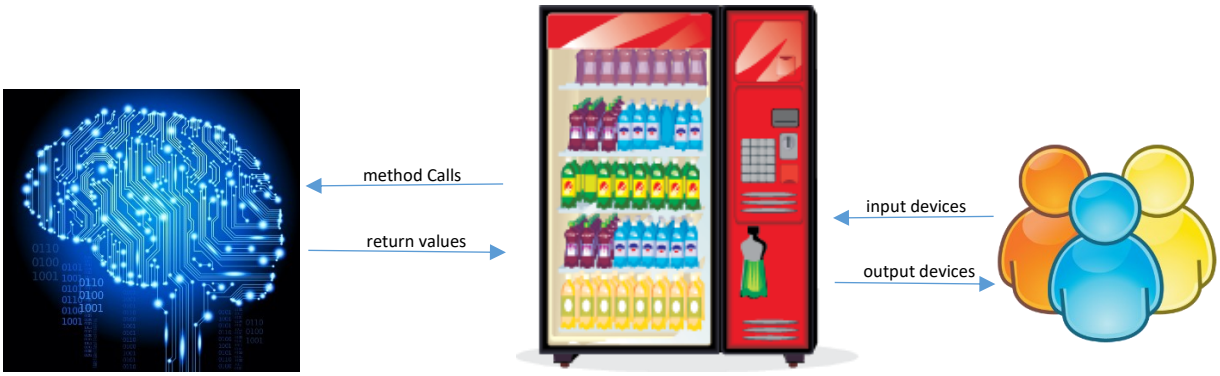


## The Assignment: Vending Machines in a Break Room

This assignment is about objects, instance variables, methods and encapsulation. A break room contains two vending machines. Your project will need two classes one called, Breakroom (which will house the main method) and one called, VendingMachine. VendingMachine is known as the **model** and Breakroom is known as the **view**.

In graphical applications, programmers often separate the **model** from the **view**. The **model** keeps track of the internal state of the vending machine, and the **view** is in the middle between the user and the **model**. The **view** talks to the user through a user interface, and the **view** also talks to the **model** by calling its methods and interpreting the return values from those methods. The model **never** talks directly to the user.

This is not a graphical application. In this assignment, the VendingMachine class is the **model** and the Breakroom class implements a "**view**" that consists of a text-based conversation with the user. But if you implement the **model** well, it should be easy to re-use it later with a new graphical **view**.




## Step 1: Design and UML

Your first step should be to create a UML class diagram to represent a vending machine. You can create this diagram using either software package, UMLet or draw.io. If you want to use a different piece of software, you must check with the instructor first.

We're going to keep things very simple. A vending machine contains a single type of product. The product has a name and price and there is a limited quantity available. A vending machine tracks its current unused credit (credit is defined to be the sum of the value of the coins inserted by the user before pressing the vend button) and the total amount of revenue the vending machine has taken in since it was switched on. The product name, price, and quantity can be set when the vending machine object is created, or the object can be created with default values for some or all these attributes.

A user can ONLY put L-loonies, T-toonies, Q-quarters, N-nickels, and D-dimes into the machine. When entering coins, use the ***first capital letter*** of the coin's name. Coins must be entered one at a time. The machine does not accept bills or pennies. When the user presses a button to vend a product the vending machine will respond with either something vended or an appropriate error response.

Vending should be denied if there is not enough credit in the machine, or if there is no product left. It is up to you how to handle the user's change. The machine might automatically return it when the vend button is pressed, or it might just deduct the price and leave it as credit. Either way, there should also be a coin return button that will return all the current credit to the user.

When the machine is first turned on, it is initialized with the name, price, and quantity of the single product it contains. There are no restrictions on how the user can use the machine. They can insert their money before they press buttons, after they press buttons, or while pressing buttons.

There should be a **toString( )** function that returns a full report, something like this:
"VendingMachine: 8 chocolate bars left, $1.99 each, $4.50 credit, balance $45.25".

**IMPORTANT BASIC RULE #1:** The vending machine is the **model** (i.e. the "brain" of an actual vending machine). The model should never talk to the user. It should do no input and produce no output. Instead, it should operate only through method calls.

## Step 2: Implement the Vending Machine and Break Room Classes

Once the VendingMachine class diagram is finished, implement your vending machine in Java code. Then write a main method in Breakroom, which will allow a user to interact with the machines in a break room. The main method should create two vending machine objects and allow a user to interact with each one using a menu interface. It's up to you how you structure this, but one possibility is shown in the example dialog below.

Sample Session

```
--------------------------------------------------------------------------------

Welcome to the Break Room!

There are two vending machines here:
1. Vending Machine:  100 Kitkat         price: $ 3.00 ea  credit: $ 0.00  revenue: $ 0.00
2. Vending Machine:  150 Hersheys       price: $ 4.50 ea  credit: $ 0.00  revenue: $ 0.00
3. Quit
Which machine do you want (e.g. 1) ? 1

What do you want to do?
1. Enter a coin
2. Coin return
3. Vend an item
4. Quit
Select (e.g. 1) > 1
Insert a coin  (e.g. L,T,Q,D,N) >  T
Your new credit is: $2.00

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

Welcome to the Break Room!

There are two vending machines here:
1. Vending Machine:  100 Kitkat         price: $ 3.00 ea  credit: $ 2.00  revenue: $ 0.00
2. Vending Machine:  150 Hersheys       price: $ 4.50 ea  credit: $ 0.00  revenue: $ 0.00
3. Quit
Which machine do you want (e.g. 1) ? 1

What do you want to do?
1. Enter a coin
2. Coin return
3. Vend an item
4. Quit
Select (e.g. 1) > 1
Insert a coin  (e.g. L,T,Q,D,N) >  2.00
```

```
Sorry, we only accept L-loonies, T-toonies, Q-quarters, D-dimes and N-nickels!
Try again!


Your new credit is: $2.00

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

Welcome to the Break Room!

There are two vending machines here:
1. Vending Machine:  100 Kitkat         price: $ 3.00 ea  credit: $ 2.00  revenue: $ 0.00
2. Vending Machine:  150 Hersheys       price: $ 4.50 ea  credit: $ 0.00  revenue: $ 0.00
3. Quit
Which machine do you want (e.g. 1) ? 1

What do you want to do?
1. Enter a coin
2. Coin return
3. Vend an item
4. Quit
Select (e.g. 1) > 1
Insert a coin  (e.g. L,T,Q,D,N) >  T
Your new credit is: $4.00

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

Welcome to the Break Room!

There are two vending machines here:
1. Vending Machine:  100 Kitkat         price: $ 3.00 ea  credit: $ 4.00  revenue: $ 0.00
2. Vending Machine:  150 Hersheys       price: $ 4.50 ea  credit: $ 0.00  revenue: $ 0.00
3. Quit
Which machine do you want (e.g. 1) ? 1

What do you want to do?
1. Enter a coin
2. Coin return
3. Vend an item
4. Quit
Select (e.g. 1) > 3
Here is your product: Kitkat


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

Welcome to the Break Room!

There are two vending machines here:
1. Vending Machine:   99 Kitkat         price: $ 3.00 ea  credit: $ 1.00  revenue: $ 3.00
2. Vending Machine:  150 Hersheys       price: $ 4.50 ea  credit: $ 0.00  revenue: $ 0.00
3. Quit
Which machine do you want (e.g. 1) ? 1

What do you want to do?
1. Enter a coin
2. Coin return
3. Vend an item
4. Quit
Select (e.g. 1) > 2
Here is your change: $1.00
-------------------------------------------------------------------------------------------

Welcome to the Break Room!

There are two vending machines here:
1. Vending Machine:   99 Kitkat         price: $ 3.00 ea  credit: $ 0.00  revenue: $ 3.00
2. Vending Machine:  150 Hersheys       price: $ 4.50 ea  credit: $ 0.00  revenue: $ 0.00
3. Quit
Which machine do you want (e.g. 1) ? 3
Goodbye!
```

**IMPORTANT BASIC RULE #2:** The main method is the **view**. It talks to the user, calls the appropriate vending machine methods in response to the user's input, and displays the results. It does not determine the results of a user's actions or keep track of money. That's the job of the vending machine objects.

> **Extra challenge:** Implement your code in the `animate` method of the `FXAnimationTemplate` instead of in a `main` method. You will still get input from the user in the console, but you can show the vending machine display on the canvas. See `GraphicsExampleForAssignment2.java` for an example of this way of using `animate`.

## Encapsulation and Documentation

Make sure you **encapsulate** your instance variables, with appropriate **get** and **set** methods and a **`toString`** method, and follow the **Documentation Standards** posted on Canvas.

### Handing In

**You have about 1 week to complete this assignment.** See the due date and time on the Canvas assignment.

Hand in the following 3 files:

A2.zip (the full IntelliJ project)
Breakroom.java.txt
VendingMachine.java.txt

### Evaluation

Your assignment will be evaluated for performance (20%), class diagram (20%), structure (40%), and documentation (20%) using the rubric in the drop box.