**Vending Machine**

**Must be completed to pass**

**Functional Requirements:**

1. Receive a budget
2. Receive item names
3. Select if item is: Clothing or Food
4. Receive item prices.
5. Calculate whether the total price of the items is within budget

**Object requirements**

* All items have –
  + Name
  + Price
* Clothing –
  + All Clothing has a Size (S, M, L, XL).
  + All clothing has a boolean Used
  + Clothing has a method Wear, which changes Used to true, if not already.
* Food –
  + All food has a **gross weight**.
  + Food has a method eat which sets the gross weight to zero when run.

**Non-Functional Requirements:**

1. Classes must be used to create the items
2. Inheritance must be used
3. The items must be stored in a way that allows the list to adjust in size for the amount of objects

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

**Optional functionality (not required to pass)**

Clothing

* Safety clothing has a **safety rating** (L, M, H)
* Casualclothing has a **style** (DayWear, NightWear, SwimWear)

Food

* Perishable items have an **expiry date**
* Non-Perishable itemshave **net weight** (the weight of the food content)

1. Use of enumerators
2. Use of getters and setters
3. Store and retrieve items using Polymorphic behaviour
4. Allow each item purchased to have a quantity associated
5. Each item may have Goods and Services Tax (GST) associated
6. Casual Clothes, Non-Perishable Foods and Other items all incur the 10% GST component (all other items must not be taxed)
7. Remove an item from the list
8. Read and Write items to a file