

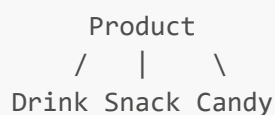
Assignment: Vending Machine Simulation Using Inheritance and Polymorphism

Assignment Instructions: Design and implement a Python program simulating a vending machine using inheritance and polymorphism. Define a base class **Product** and derive multiple product types from it, each overriding relevant methods.

Class Design:

- Base Class: **Product**
 - Attributes: **name**, **price**
 - Method: **display_info()** – displays basic product details.
- Subclasses:
 - **Drink**: adds attribute **volume**, overrides **display_info()**
 - **Snack**: adds attribute **calories**, overrides **display_info()**
 - **Candy**: adds attribute **flavor**, overrides **display_info()**

UML Diagram:



Runtime Behavior Example:

Extended Feature – Loading Products from a File: To simulate a realistic vending machine, extend the program to load products from a file.

Example File Structure (**products.txt**):

```
Drink,Cola,1.50,500
Snack,Chips,2.00,250
Candy,Gummy Bears,1.20,Strawberry
Drink,Water,1.00,600
Snack,Cookies,1.75,300
```

Each line contains: **Type**,**Name**,**Price**,**Attribute**

- For **Drink**: the last value is volume in ml
- For **Snack**: the last value is calories
- For **Candy**: the last value is flavor

Student Task:

- Parse this file line by line
- Create the correct object (**Drink**, **Snack**, or **Candy**) based on the type

- Store all objects in a list
- Show a menu based on the loaded products

Expected Menu Output:

```
Welcome to the Python Vending Machine!

Please select what you want:
1. Drink - Cola
2. Snack - Chips
3. Candy - Gummy Bears
4. Drink - Water
5. Snack - Cookies

> 1

Product Information:
Product: Cola, Price: $1.50
Volume: 500ml
```

Hint for Students: Use `open()`, `readlines()` or `csv.reader` to process the file. Implement a simple factory method or conditional logic to instantiate the correct subclass.

```
Welcome to the Python Vending Machine!

Please select what you want:
1. Drink
2. Candy
3. Snack

> 1

Product Information:
Product: Cola, Price: $1.50
Volume: 500ml
```

Note: You must:

- Use inheritance and polymorphism.
- Use `super()` to reuse the base class methods.
- Override `display_info()` in each subclass.
- Present the user with a menu and respond accordingly.