Capstone 1: Vending Machine

Team 2: Andrew & Rachel

Category	Feature	Score	Notes
Features	Requirements	3	• All requirements were met! • Select Product should list the products for me to choose from • I lost a dime. When my balance was 0.95, you gave me back 3 quarters and a dime.
	Program startup VM Creation and load Sharing of VM	3	• Program creates a VM and then passes it into the Menu constructor. Good. • CSV file is hard-coded into the VM, making it impossible to stock other items or even to test other items being stocked.
	How change is made	2	• The logic is fine (except for the bug), but the response should not be output to the user. This cannot be tested.
	How is product selected and dispensed	1	• All of this logic is in the switch statement in the menu. You should consider how you might re-factor this.
Architecture	Use of OO techniques	1	• At the very least, the consumption message should be encapsulated in the Slot class, not the menu. • There is very little encapsulation. Vending Machine's balance and inventory are both public get/set. The VM and Menu classes are very tightly coupled.
	Error Handling		• Since the VM is doing almost nothing, there are few errors to be thrown or caught.
Maintainability	Code comments	0	• There are no comments anywhere in the code
	Testability of code	0	• All significant logic is in the menu, and since this is where the user I/O is, is not testable. Vending Machine should have methods for selecting and dispensing product, finishing the transaction and making change.
	Tests	1	• The only tests possible was to feed money and check the balance. This is a small and fairly trivial part of the overall logic.