

Capstone 1: Vending Machine

Team 5: Daniel & Joan

Category	Feature	Score	Notes
Features	Requirements	3	<ul style="list-style-type: none">• All requirements were met! Great job!• Nice graphic.
	Program startup VM Creation and load Sharing of VM	3	<ul style="list-style-type: none">• Program creates a VM and passes it into a new Menu. Great.• VM stocks itself from its constructor, and uses a hard-coded file path. This makes it hard to change stock sources, and difficult to test in different configurations. It least a path should be passed in, if not having a completely separate loader and passing in a list of items.
	How change is made	3	<ul style="list-style-type: none">• VM does the work, and returns a Change object. Love it!
	How is product selected and dispensed	3	<ul style="list-style-type: none">• All the validation and dispense work is handled in the vending machine, with no user I/O. This makes the machine very testable.
Architecture	Use of OO techniques	2	<ul style="list-style-type: none">• VM should be encapsulated more. Balance, TotalSales, items, should all be a least private set. Outsiders should not be able to mess with them.• Dispense message is handled with a dictionary on the VM. This should probably come from the item class itself.
	Error Handling	3	<ul style="list-style-type: none">• VM throws custom exceptions, menus catch and communicate with the user. Great!
Maintainability			
	Code comments	3	<ul style="list-style-type: none">• The best-commented capstone I have seen so far.
	Testability of code	3	<ul style="list-style-type: none">• Well separated logic, should be very testable.
	Tests	3	<ul style="list-style-type: none">• Good selection of tests• Would be nice to load something other than the standard CSV that came with the project, for the tests.