CS 449 Homework 2

Kory Overbay

1. The item class was created with the bare requirements in its own package. It’s covered within the sale tests file as it’s used in each function. The test coverage includes happy path tests that the functions are working as well as tests ensuring that subtotals and discounts cannot go negative or behave in an unexpected manner.
2. For the new requirements the Sale class was updated to have a new string variable for customer type. The getDiscountRate function was modified to have a switch statement that compares the customer type to constant strings Senior and Preferred to determine if a discount is to be added to the existing 0.05 rate. The test cases were modified by adding 3 sale instances for each customer type, and adding the same data to all three and asserting expected results.
3. For the new requirement, an if statement was included in the switch case Senior condition to consider if the sale’s day was Tuesday or not and returns the appropriate amount. The getDiscountRate and getDiscountedSubTotal tests were modified to include a sale version of senior where it is and is not Tuesday. The getSubTotal tests was not modified because it alone is unaffected by the day of the week for seniors.
4. The class hierarchy was changed in a way that a switch case was not needed to determine the type of customer. This was achieved through inheritance across the Sale and Customer classes and polymorphism in the abstract getDiscountRate function that behaves differently based on the customer class calling it. Tests were updated to use the customer classes instead of the Sale class.
5. The item class was given a specialItemCode integer to make certain items distinct from others (defaults to 0). The sale class now checks for the item code and then discounts multiple items accordingly. The tests were modified when the second item is added to the sale, this time with a quantity > 1 and asserts that the correct discounts and totals are calculated.