

## Integration Tests

### **Test Case:** IntegrationTest 1

#### **Test Name:** SalesNumber getSalesInfo() Integration Test

**Description:** Testing the getSalesInfo() function will ensure that administrative users are able to access the sales analytics for their respective stores in the system. The method retrieves the information from each store's sales analytics, containing vital information such as an item's unique ID, the quantity of said items sold, and the item's Name. This information is used when deciding whether to or whether to not order more stock of an item given the demand from customers. This information is also used to see whether a store location is thriving in business, or vice versa given the quantity of sales per location. This function is located in the Sales Number class, can be accessed from the Administrative User class, and accesses the Transaction History class and Inventory class. The Sales Number class accesses the Inventory class to update the inventory stock with the quantity of items sold, and uses the Transaction History class to know the quantity of items sold, alongside which items are being sold.

**Steps:** Using the setter function in the SalesNumber class, set each variable to a desired value. Create a tester class with SalesNumber object x. Using this object, call the following functions with a test set of inputs: x.setSalesName("Slacks"), x.setSalesQuantity(27), x.setSalesIdNumber(0123456). After calling these setters, the result should be printed to x.getSalesInfo(), which returns as a string containing the resulting sales attributes.

**Input(s):** x.setSalesName("Slacks"), x.setSalesQuantity(27), x.setSalesIdNumber(0123456)

**Expected Result(s):** The result of calling the getSalesInfo() function should return a string that outputs: "Item Name: Slacks, Item Quantity Sold: 27, Sales ID: 0123456"

**Actual Result(s):** "Item Name: Slacks, Item Quantity Sold: 27, Sales ID: 0123456"

### **Test Case:** IntegrationTest 2

#### **Test Name:** Payment getPayment() Integration Test

**Description:** Testing the Payment class' getPayment() function will check if the customer's transaction information is stored properly within the Transaction History class. This function references the Credit Card, Debit Card, or Cash class. By using customer information within the aforementioned classes, it stores thecustomerPaymentType variable, and the customerCardInfo variable in the Transaction History class. This method is important as refunds through credit and debit cards are processed through the customer's card information that's stored in the Transaction History class, making the function a cross-class method.

**Steps:** Using the setter functions in the Payment class, the paymentInfo variable will contain the customer's credit card or debit card information. We'll be testing if the input passed onto the setter class in the Payment class will properly transfer over into the Transaction class' getter class. Our first step is to call setPaymentInfo(3300815134028233). This is then printed into transactionHistory.getPaymentInfo() as an int of the customer's 16-digit credit card number.

**Input(s):** 3300815134028233 is inputted into setPaymentInfo(3300815134028233)

**Expected Output(s):** 3300815134028233

**Actual Output(s):** 3300815134028233