**Assignment: 2**

**Date of Submission: 18th April, 2018**

**CoffeeMaker** is a coffee machine simulator written in java . The CoffeeMaker performs various functions. The detailed specification of the acceptance tests are following:

AddInventory Acceptance Test

Inventory may be added to the machine at any time. The types of inventory in the Coffee Maker are coffee, milk, sugar, and chocolate. The inventory is measured in units. No inventory may be taken away from the CoffeeMaker.

AddRecipe Acceptance Test

Only three recipes may be added to the CoffeeMaker. A recipe consists of a name, price, units of coffee, units of milk, units of sugar, and units of chocolate. Each recipe name must be unique in the recipe list. Price must be handled as an integer.

CheckInventory Acceptance Test

Inventory may be checked at any time. The units of each item in the inventory are displayed.

DeleteRecipe Acceptance Test

A recipe may be deleted from the CoffeeMaker if it exists in the list of recipes in the CoffeeMaker. The recipe to be deleted is chosen by its name. User will be given a warning before deleting the recipe.

EditRecipe Acceptance Test

A recipe may be edited in the CoffeeMaker. The user will be prompted for the name of which recipe they wish to edit, then they will be asked to reenter the recipe information. A recipe name may not be changed to a name of any other recipe in the recipe list.

PurchaseBeverage Acceptance Test

The user will not be able to purchase a beverage if they do not deposit enough money into the CoffeeMaker. Change will be dispensed to the user, if they paid more than the price of the beverage. A user's money will be returned if there is not enough inventory to make the beverage.

**Assignment Tasks**

1. For the above specifications design test cases by performing following tasks:

a. Identify a set of features which if successfully tested can verify all the acceptance tests.

b. For each feature identified above

Find input data that tests different aspects of feature.

Find corresponding output data that verifies a correct behavior.

2. Perform the above tests using **JUnit** and report the following:

a. Each Test Case and its results.

b. The number of failures that occurred.

c. A failure list, which briefly describes the test case, the output and expected output.