

## 19.02 Assignment Instructions

**Instructions:** For this assignment, you are going to apply your knowledge of assertions and exceptions.

1. Create a folder called **19.02 Assignment** in your module 19 assignments folder.
2. You are to create a class called **Dispenser** and save it as **Dispenser.java**.
  - a. **Dispenser** should have instance variables **numberOfItem**, and **cost** (of type **int**).
  - b. The default constructor should set **numberOfItems** and **cost** to 50.
  - c. A second constructor should accept values for **numberOfItems** and **cost**. It should test to make sure that no bad values get assigned, and appropriately throw exceptions.
  - d. Remaining methods will be **getCount()** – returns **numberOfItems**, **getProductCost()** – returns **cost**, and **makeSale()** that subtracts one from **numberOfItems**.
3. You are to create a class called **CashRegister** and save it as **CashRegister.java**.
  - a. **CashRegister** should have instance variable **cashOnHand** of type **int**.
  - b. The default constructor should set **cashOnHand** to 500.
  - c. A second constructor should accept values for **cashOnHand**. It should test to make sure that no bad values get assigned and appropriately throw exceptions.
  - d. It should have a method **CashRegister** that accepts an **int** value and assigns it to **cashOnHand** only if it is greater than zero. It should throw an exception as needed.
  - e. It should have a method **acceptAmount** that accepts an **int** value and adds it to **cashOnHand**, as long as it is greater than zero. It should throw an exception as needed.
4. You are to create a class called **CandyMachine** and save it as **CandyMachine.java**.
  - a. This class will test your other classes. You should test your classes to make sure they appropriately throw exceptions.
  - b. Create method **sellProduct** that accepts a dispenser, cashregister, and amount inserted as arguments. It should then sell product if it can or display a message if it cannot.
  - c. Output will be similar to that shown below, but, if tested with a candy dispenser (set up with price of 50 cents, and quantity 100) and coins of 75, output would be:

