Lab 13
Apple Maker

Objective:

Write a program that creates a class **Apple** and a tester to make sure the Apple class is crisp and delicious.

First create a class called **Apple**

- Write a class file called **Apple** that DOES NOT HAVE a main method
- Some of the attributes of Apple are
  - Type: A string that describes the apple. It may only be of these following types
    - Red Delicious
    - Golden Delicious
    - Gala
    - Granny Smith
  - Weight: A decimal value representing the apple's weight in kilograms. The weight must be between 0kg and 2kg
  - Price: The price per apple. This must be a non-negative decimal value.
- Create the following Constructors
  - Default – sets everything to default values and has no parameters
- Accessors and Mutators for each variable
  - MAKE SURE THE MUTATORS CHECK FOR VALID VALUES!
- Create the following Methods
  - toString: Returns a string with all of the instance variable values
    - Ex: "Name: <<apple's name>> Weight <<apple's weight>> Price <<apple's price>>"
  - equals: This method returns a Boolean and takes in another instance of an apple. It returns true if all of the instance variables equal the instance variables in the other apple

Finally create a class called **AppleTester**

- This class DOES HAVE a main method
- Create at least 3 different types of apples
- Test if the accessors, mutators, and other methods work as intended.

Example Dialog:

```
Welcome to the apple tester

Creating a default apple
Printing the default apple's values
Name: Gala Weight: 0.5 Price: 0.89

Creating another apple
Setting the new apple's values to the following valid values "Granny
Smith" 0.75 0.99
Printing the new apple's values
Name: Granny Smith Weight: 0.75 Price:0.99

Creating another apple
Setting the new apple's values to the following invalid values "iPad"
2.5 -200
Invalid Name
Invalid Weight
Invalid Price
Printing the apple's values which should have not changed from the
default values
Name: Gala Weight: 0.5 Price 0.89
```

Lab Report Questions:

1. In your own words describe what a class is used for.
2. In your own words describe encapsulation as it relates to object oriented programming

Finally:
Upload ALL java files to the dropbox