# CS-207: Programming II
# Spring 2016
# Northeastern Illinois University
# Homework #5: Due 02/25/16 at 9:00 a.m.
# Relationships, Wrappers and BigInteger/BigDecimal

## Problem #1

Create the following three classes, `GroceryList`, `Item`, and `GroceryTest` according to the specifications below. As you begin this problem, ask yourself the the following questions:

1. Which class do I create first? Why?
2. What type of relationship do the `GroceryList` and `Item` classes have with each other?

### Specifications for the `GroceryList` class:

- Make sure your class is correctly encapsulated and use the keyword `this` with **ALL** instance variable usages.
- An instance variable named `groceries` that is an `Item`-type 1D array (i.e. `Item[]`).
- A constructor that takes one parameter that is of type `Item` and initializes the instance variable to be an array of length one that contains that `Item`.
- A getter method for the instance variable.
- A public method named `addItem` that takes one parameter of type `Item` and modifies the instance variable so that the item is appended to the end of the list (Hint: You may need to create a new 1D array with the correct dimensions and reassign it to your instance variable). This method should not return anything.
- A public method named `removeItem` that takes one parameter, an `int`, and modifies the instance variable so that the item at the specified parameter is removed (Hint: You may need to create a new 1D array with the correct dimensions and reassign it to your instance variable). This method should not return anything.

### Specifications for the `Item` class:

- Make sure your class is correctly encapsulated and use the keyword `this` with **ALL** instance variable usages.
- A String instance variable named `name`.
- A double instance variable named `price`.
- A constructor that takes two parameters, a String and a double, and sets the instance variables.
- A public method named `display` takes no parameters and does note return anything. The method should print out the value of the `name` instance variable followed by a space and then the value of the `price` instance variable.

### Specifications for the `GroceryTest` class:

- The `GroceryTest` class should contain the `main` method.
- You can use reference variable names of your choice.
- Create a new `Item` object that has the name `"Bananas"` and a price of 0.99 and assign it to a reference variable.
- Create a new `GroceryList` object and pass in the `Item` reference variable that you just created.
- Create two new `Item` objects (and corresponding reference variables). One `Item` should have a name of `"Saltines"` and a price of 1.23 and the other should have a name of `"Chocolate"` and a price of 4.57.
- Add each of the new `Items` to your grocery list.
- Use the getter to get the array of `Items` and assign it to a reference variable.
- Iterate over each element in the array of `Items` and use the `display` method to print out the name and price of each `Item`.
- Remove the `Item` at index 1 in your grocery list.
- Use the getter to get the array of `Items` and assign it to a reference variable.
- Iterate over each element in the array of `Items` and use the `display` method to print out the name and price of each `Item`.
- If you created your classes (including the test class) correctly, you should see the output below.
- Place all three of your files in a folder named Homework5 to be submitted to D2L

```
Bananas 0.99
Saltines 1.23
Chocolate 4.57
Bananas 0.99
Chocolate 4.57
```

### Problem #2

Create a class named `StringParser` that has the following:

- A public static method named `findIntegerDivisors` that takes a `String` and two `char` variables as parameters (in that order) and does not return anything.
- The method should find the integer value that is located in between the two characters.
- The method should then print out all of the numbers between 1 and the integer (inclusive) that divide evenly into the integer on the same line separated by spaces.
- If no number appears in between the specified `char` parameters, print out `"No integer found."`.

- You can assume that only valid integers or nothing will appear in between the specified characters and that each string will only contain one integer in between the characters. In addition, you can assume that the two `char` parameters will each only appear once in the `String` parameter, with the second `char` always following the first `char` parameter.
- Download the `StringParserTest` from the NeededFiles.zip file and compile and run it.
- If you created your class and method correctly, you will see the output below.
- Place your StringParser file into the Homework5 folder to be submitted to D2L.

```
String is: rugtsbckgus!32*
1 2 4 8 16 32
String is: disdkfjs<873>sfjsldkfiwx
1 3 9 97 291 873
String is: rujfbgl&%fkslga
No integer found.
```

**Problem #3**
Create a class named `CircleArea` that has the following:

- A public static method named `findArea` that takes a double as a parameter representing the radius of a circle.
- The method should calculate the area of a circle using the formula: $A = \pi r^2$, where $\pi = 1.314159$
- The method should return the value of the area (which should be of type `BigDecimal`).
- The method should work for very large numbers.
- You may find the following link to the JavaDocs helpful: `https://docs.oracle.com/javase/8/docs/api/java/math/BigDecimal.html` Please note that the `BigDecimal` class has many more constructors than the `BigInteger` class. In fact, it has a constructor that takes a double primitive type as a parameter! Similar to the `BigInteger` class, the `BigDecimal` class also has methods for multiplying and raising a number to a power.
- Download the `CircleAreaTest` from the NeededFiles.zip file and compile and run it.
- If you created your class and method correctly, you will see the output below.
- Place your `CircleArea` file into the Homework5 folder to be submitted to D2L.

```
Area of a circle with radius 4.5 is:
63.6171974999999976230213860617368482053279876708984375
Area of a circle with radius 3.704383383884849E175 is:
4311033134552392262055514146377395282591560572954698770240989943
8478574163766585430010996238451781484279877853786364232253323833
2206090247513208365107366111335629644512872900093516275861023839
8897497086633104024831201183171779899926117040141514978190169025
5955112809797340642003707810062075175422647160890029042854800144
77091862687460167071749399117824.0000000000000000000000000000000
0000000000000000000
```

**A note on cheating/plagiarism:**
A plagiarism detector is used on all submitted code (across all sections) for homework assignments. If the plagiarism detector determines that 25% or more of your code for a particular assignment is plagiarized, you will receive a zero (i.e. an F) for that homework assignment, regardless of whether you cheated from someone or vice-versa. If you plagiarize half or more of the total homework assignments, you will receive a zero for the entire homework percentage.

**Submitting your assignment to D2L**

1. Make sure your name and assignment number are in the .java file(s) (as comments) and text file.

2. Place all your files in a folder and compress (i.e. .zip) the folder. Submit the .zip file to the Homework #4 folder on D2L. You should submit only one file - the .zip file. Do **NOT** upload multiple files.

3. Turn your homework in to D2L by the specified deadline (no late homework will be accepted - see syllabus for policies)