

Assignment Prefix: Lab07

Due Date: Friday, March 3rd @ 11:59pm

Points: 100

This is an individual assignment.

Restrictions: you cannot use any methods from the Java Array(s) class to copy an array, check for equality, or otherwise manipulate an array. You must write the Java code to perform these functions.

The above restrictions do not apply to any of the classes copied from the textbook.

Create a NetBeans project named Lab07 and ensure it is saved to a location like desktop or your flash drive. In the project you will do the following:

For this assignment you are going to explore the wonderful world of iterators.

Step 1:

Create a class named **LuckyNumber** that has the following instance variables:

- String name
- int luckyNumber

This class should **only have** a single overload constructor that:

- takes a single parameter, name
- automatically assigns luckyNumber a random number between 0 and 9 (inclusive of both) to the instance being created

This class **must have** the following methods:

- getName
- get luckyNumber
- toString
 - o make sure your toString returns the class name
 - o the class name will be useful if you “become lost” in your data structure
- equals

This class **should not have** the following methods:

- set name
 - o you are who you are
- set luckyNumber
 - o you can't change your luck

Step 2:

Create a class named **LuckyNumberList** that will contain a `LinkedPositionalList` of `LuckyNumber` objects.

This class should contain a constructor and methods similar to the `Alphabet` class in the `IteratorExample1` project.

- Unlike the `Alphabet` class your constructor should just create an empty `LinkedPositionalList`
- You will need to add an **addLuckyNumber** method to this class
 - o This method has a single parameter of type `LuckyNumber`.
 - o this method should add the new entry at the end of the list.
- It will be the job of the client class the add entries into the list.

You will be adding two custom iterators:

- an iterator that iterates over the positions in the list that have an even `LuckyNumber`
- an iterator that iterates over the positions in the list that have a prime `LuckyNumber`
- Note, if you use the `Alphabet` class as a guide you will remove the `VowelPositionIterator`. Actually, you may find it helpful to modify the `VowelPositionIterator` into one of the above required iterators.

Step 3:

Create a client class named **Client** that does the following:

- Creates an instance of your `LuckyNumberList` class.
- Fills the list with at least ten (10) names.
 - o You can hardwire the fills into your code by using ten calls to `addLuckyNumber` that pass new instances of `LuckyNumber` objects.
 - o Names can be hardwired into your code.
- Prints out the resulting contents of your instance of the `LuckyNumberList` class using:
 - o The default iterator that displays all entries in the list
 - o The Prime iterator that displays only the entries in the list that have prime `luckyNumbers`
 - o The Even iterator that display only the entries in the list that have even `luckyNumbers`.

- You output should be similar to the output listed below:
 - o Correct use of the printf method makes printing this output easy
 - o Your output should be nicely formatted
 - o You may need to run your program several times in order to get a good mix of even/odd and prime/not prime entries.

Lucky Number List Contents (printed with default iterator)

Joe	0	Even	Not Prime
Tina	8	Even	Not Prime
George	2	Even	Prime
Bobbie	5	Odd	Prime
Fred	1	Odd	Not Prime
Karen	9	Odd	Not Prime
Linda	0	Even	Not Prime
Gale	3	Odd	Prime
Rick	0	Even	Not Prime
Sharon	6	Even	Not Prime
Frankie	5	Odd	Prime
Vance	2	Even	Prime
Lori	0	Even	Not Prime

Lucky Number List Contents (printed with Prime iterator)

George	2	Even	Prime
Bobbie	5	Odd	Prime
Gale	3	Odd	Prime
Frankie	5	Odd	Prime
Vance	2	Even	Prime

Lucky Number List Contents (printed with even iterator)

Joe	0	Even	Not Prime
Tina	8	Even	Not Prime
George	2	Even	Prime
Linda	0	Even	Not Prime
Rick	0	Even	Not Prime
Sharon	6	Even	Not Prime
Vance	2	Even	Prime
Lori	0	Even	Not Prime

Things to turn in:

- Copy and Paste the source code of the Client (make sure to use *Ctrl + A* to select all the source code of the program and *Ctrl + C* to copy).
- You **DO NOT** need to include copies of any interfaces or classes that you copied out of the textbook but did not modify/customize.
- You **DO** need to include the classes that you modified.
- Copy and paste the output of the client program
 - Create a screen capture of your NetBeans IDE that includes the contents of the Output Window and paste it into your Word document below your source code.
 - To create a screen capture of your NetBeans IDE
 - Select, left-click in the NetBeans IDE
 - Use Alt-PrintScreen to place an screen capture image on the clipboard
 - Use Ctrl-V to paste the contents of the clipboard into your Word document
- Next, zip the Project folder.
- Finally on blackboard, submit both your Word document and project zipped file.