CSE105 Labs Week 10

There are two parts to this lab:

- a checkpoint on ArrayList and Iterator
- a practice task for CW2

Part 1 Checkpoint

1. Create a new project. In the main class create an ArrayList of type String called *names*. It should be a static Class variable (class scope: can be used anywhere in the class).
2. In the main method, add at least five different names (Strings) to the ArrayList.
3. Create a new static method called removeString.
    a. It should take a String as an argument, and return a boolean.
    b. It should search the names ArrayList to see if it contains the String argument.
    c. If it does, it should remove the String from the ArrayList and return true.
    d. If the ArrayList does not contain the String, it should do nothing and return false.
4. Call the removeString method at least twice from main() and print the result to the console.
5. Finally, loop through the *names* ArrayList and print it to the console.

When you are happy with this: Checkpoint!

Note: if all the TAs are busy, get on with part 2 and checkpoint later.

Part 2: CW2 Practice task

ISBN Validation

It is very useful to write programs that can validate user input. This means that we can format initial input and check it, before carrying out any further processing. This can stop errors from happening later in more complex programs. An ISBN is a unique code that is applied to all books.

Actual sample ISBNs:

ISBN 978-0-273-72131-4
ISBN 978-3-540-74549-5

- Download the ValidateTaskExample.java file and use it to write a program that accepts as input an ISBN. The program validates that it is an ISBN and will return a boolean value to indicate if it is a valid number

- A correct ISBN will:
    - Begin with “ISBN” (lower case and upper case are both fine.)
    - One space ‘ ’ after “ISBN”
    - Contain only digits (0 to 9) and “-“.
    - The highlighted positions have to be “-”
    - Be 22 characters in length

- The validation MUST occur in the validateString method, receive a string, and return a boolean variable.

- You should have a test String for each possible error that you validate. You should test your validateString method, using these Strings, from the main() method.

- Thereturned boolean should be used to print a message to show if the ISBN is valid or not for your test.

- This is a challenging problem, so think about how you would solve it. There are many ways you can do it. Think about the steps needed to solve this problem, based on the requirements. Design your solution first and then implement it.

- Remember, there are MANY different ways to solve it. Some possible tips (do **not** try them all!):
    - First, and most obvious, check the length of the string
    - Check that your substrings match a specified format (i.e. “ISBN”) or hyphens in the right place
    - Consider groups of characters as substrings, and evaluate them separately
    - Check that your value is a number by parsing it to an integer or long, and handling any exception thrown.
    - Or You could try converting your input string to individual characters and then checking that the character is between 0 and 9

- Your Result should look similar to:

```
==============================
"ISCN-978-0-273-72131-4"
It does not start with ISBN.  false
==============================
"ISBN 97a-0-273-72131-4"
digits are wrong.  false
==============================
"ISBN 978-b-273-72131-4"
digits are wrong.  false
==============================
"ISBN 978-0-2v3-72131-4"
digits are wrong.  false
==============================
"ISBN 978-0-273-7d131-4"
digits are wrong.  false
```