Java Programming
CSPC 23
Assignment 4

1. In an array list of strings, make each string uppercase. Do this with the following:
   a. An iterator
   b. A loop over the index values
   c. Using the ***replaceAll*** method

2. Given two stacks of textbooks of the following subjects **"Chemistry"**, **"Mathematics"**, **"Biology"**, **"English"** and **"Biology"**, **"English"**, **"Geography"**, **"Physics"** respectively; find the subjects that are:
   a. Only present in the first stack
   b. Only present in the second stack
   c. Present in both stacks.

   (You may clone the sets to preserve the original sets from being changed by set methods.)

3. Given one or more text files, each representing a day's attendance in a course and containing the names of the students who attended the course on that particular day, write a program that displays, in ascending order, the names of those students who have attended at least one day of the course. The text file(s) is/are passed as command-line argument(s).

4. Write a program that prompts the user to enter a text file name and displays the number of vowels and consonants in the file. Use a set to store the vowels A, E, I, O, and U.

5. Write a program that prompts for and reads a course name, its credits and reference book. Then print the following paragraph, inserting the appropriate data:

   `This semester, a new course on` ***`course_name`*** `has been added to the curriculum. It consists of` ***`credits`*** `credits and the reference book for this course is` ***`reference_book`*** **`.`**

6. Write a program that prompts for and reads a double value representing a monetary amount. Then determine the fewest number of each bill and coin needed to represent that amount, starting with the highest (assume that a ten-dollar bill is the maximum size needed). For example, if the value entered is 47.63 (forty seven dollars and sixty-three cents), then the program should print the equivalent amount as:

```
4 ten dollar bills
1 five dollar bills
2 one dollar bills
2 quarters
1 dimes
0 nickles
3 pennies
```

7. Write a class called ***Phone*** that contains instance data that represents the make, model, and year of the phone. Define the ***Phone*** constructor to initialize these values. Include getter and setter methods for all instance data, and a ***toString*** method that returns a one-line description of the phone. Add a method called ***isObsolete*** that returns a boolean indicating if the phone is obsolete (if it is more than 10 years old). Create a driver class called ***PhoneCheck***, whose main method instantiates and updates several Phone objects.