American University of Beirut
Department of Computer Science
CMPS 200 – Introduction to Programming
Assignment 8: April 28 @ 11:00 PM

Exercise 1. Max Length

Write a method maxLength that takes an ArrayList of Strings as a parameter and that returns the length of the longest string in the list. If your method is passed an empty list, it should return 0.

Exercise 2. Repeated Strings

Write a method **repeat** that takes an ArrayListofStrings and an integer k as parameters and that replaces every string with k copies of that string. For example, if the list stores the values["how", "are", "you?"] before the method is called and k is 4, it should store the values ["how", "how", "how", "how", "are", "are", "are", "are", "you?", "you?", "you?", "you?"] after the method finishes executing. If k is 0 or negative, the list should be empty after the call.

Exercise 3. Hangman (Bonus)

Write a program `Hangman.java` that emulates the game Hangman. The game is played as follows: to win the game, the user must guess a word letter-by-letter before exhausting the maximum number of guesses;

The program must read the words to be guessed from the input file `words.txt`. It must also ask the user to enter the maximum number of guesses N allowed. The program must also keep track of the guessed and missed letters, and should discard already attempted guesses (and misses). If the user guesses a letter that is repeated in the word, all the (same) letters must be revealed to the user, and not just one. *PS*: when comparing letters, you should ignore case. Example:

If the word is "test", a winning scenario:	If the word is "test", a *losing scenario*:

`Enter the maximum number of guesses: 6`
`Word: _ _ _ _`
`Misses:`
`Guess: r`
 	
`Word: _ _ _ _`
`Misses: r`
`Guess: t`
 	
`Word: t _ _ t`
`Misses: r`
`Guess: g`
 	
`Word: t _ _ t`
`Misses: r g`
`Guess: e`
 	
`Word: t e _ t`
`Misses: r g`
`Guess: e`
 	
`Word: t e _ t`
`Misses: r g`
`Guess: s`
 	
`You Won! The word was: test`
	`Enter the maximum number of guesses: 3`
	
	`Word: _ _ _ _`
	`Misses:`
	`Guess: r`
	
	`Word: _ _ _ _`
	`Misses: r`
	`Guess: T`
	
	`Word: t _ _ t`
	`Misses: r`
	`Guess: r`
	
	`Word: t _ _ t`
	`Misses: r`
	`Guess: p`
	
	`Word: t _ _ t`
	`Misses: r p`
	`Guess: g`
	
	`You Lost! The word was: test`

Exercise 4. Complex Numbers

This problem revolves around the creation of a class for representing complex numbers. Prior to diving into the details of this classwork, below is a premier on complex numbers. A complex number z is such that:

z = x + iy (1)

where x, y and z are referred to as the **real part**, **imaginary part** and **imaginary number** with $i^2 = -1$ or equivalently $i = \sqrt{-1}$.

The above Equation (1) is known to be the cartesian representation of the complex number z. Now, z can be interpreted as a vector in the complex plane where x is the component of that vector along the Reals axis (i.e. Re) and y is the component of that vector along the Imaginaries (i.e. Im) axis as shown in Figure 1:

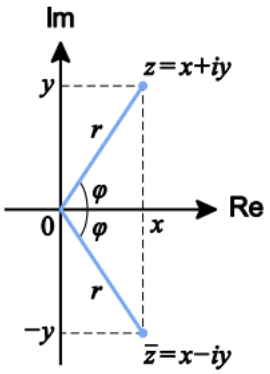

Figure 1

Such a vector has a **length**, otherwise referred to as the **magnitude of z** and computed as:

$$|z| = \sqrt{x^2 + y^2} \quad (2)$$

Also, the **angle** between that vector and the Reals axis is referred to as the **argument** or the **phase angle** of z and is computed as:

$$\varphi = \tan^{-1}(y/x) \quad (3)$$

The **reflection** of the vector represented by z with respect to the Reals axis is another vector having the same magnitude but a phase angle of $-\varphi$. This latter vector is represented by a complex number:

$$\bar{z} = x - iy$$

which is referred to as the **complex conjugate of z**. The real part of $\bar{z}$ is x (i.e. the same as the real part of z) while its imaginary part is $-y$.

Complex numbers, similar to regular arithmetic, may be added, subtracted, multiplied and divided. For example, consider two complex numbers $z_1 = x_1 + iy_1$ and $z_2 = x_2 + iy_2$, therefore:

$$z_1 \pm z_2 = x_1 + iy_1 \pm x_2 \pm iy_2 = (x_1 \pm x_2) + i(y_1 \pm y_2) \quad (4)$$

$z_1 \cdot z_2 = (x_1 + iy_1)(x_2 + i\,y_2) = x_1\,x_2 + i\,x_1\,y_2 + i\,x_2\,y_1 + i^2\,y_1\,y_2 = (x_1\,x_2 - y_1\,y_2) + i(x_1\,y_2 + x_2\,y_1)$ (5)

$$\frac{z_1}{z_2} = \left(\frac{x_1x_2+y_1y_2}{x_2^2+y_2^2}\right) - i\,\left(\frac{x_1y_2+x_2y_1}{x_2^2+y_2^2}\right) \quad (6)$$

Write a JAVA class called ComplexNumber that represents a complex number z. This class, must include all variable and procedural attributes.

1. An object of type ComplexNumber has **two variable attributes**: **real** and **imaginary** parts.

2. Procedural attributes must incorporate:
   - the **constructor** method
   - an appropriate set of **accessor** methods (getters)
   - an appropriate set of **mutator** (setters)
   - operational methods allowing the performance of the above listed operations in equations (4), (5) and (6)
     - Note that, upon performing an operation involving two complex numbers, none of these numbers' internal variable attributes is allowed to be changed (i.e. the values of the internal variable attributes must be preserved allowing that same object to be re-used in other operations).
   - accessor methods must also include methods to return the **magnitude**, **phase** angle in degrees and **complex conjugate**.
   - Finally, an appropriate **toString**() method must be defined to display the complex number in the form x + iy (if y > 0) and x – iy (if y < 0).

     If the complex number has an imaginary part which is equal to 0, then only the real part must be displayed.

     Also, if the complex number has a real part which is equal to 0 and non-zero imaginary part, only the imaginary part must be displayed as iy (if y > 0) and –iy (if y < 0).

     If both real and imaginary parts are 0, then only 0 must be displayed as a result of printing the complex number to the screen.

     Now, if the imaginary part is equal to 1, then the complex number must be displayed as x + i (not as x + i1).

     Similarly, if the imaginary part is -1, then the complex number must be displayed as x - i (not as x - i1).

Write a JAVA client program called ComplexClient.java to test and show the proper functionality of the above custom-defined class.