Working with classes and methods

Some keywords to remember: Class, object, method,..

Class as a type

- A class is like a blueprint from which an object is created.
- We can create many objects from the class.
- The differences among these objects are the attribute values (data) that define the objects' state.
- For example, a class **Student** might be used to create a student1 object.

Objects:

- An object has both state (properties) and behaviors (methods).
- An object's current state (data) is defined by the values for its attributes. These values are stored internally and may require a little or a lot of memory depending of theirs sizes, the default values for numbers data type is 0.
- An object's behaviors (methods) are the actions it can perform.
- The type (or category) of an object is its class.
- Java has many classes already defined: String, System, Scanner. (Recall: Class names start with a capital letter.)

Object Behaviors

- To use an object, we need to know only the behaviors of an object.
- An object's behaviors are defined by a set of methods associated with the object.
- For example, methods may enable you to access or change an object's attribute values, or to ask the object to perform a task.
- These methods are known as the interface to the object.

String Objects

- An object of type String holds a sequence of (unicode) characters.
- When we declare a variable of type String, it does not create an object. All you get is a way to refer to the object: **String** name; //name is the object
- To create an object we use the new operator: **String** name1 = **new** String("John");
- Strings have a shortcut way of creating them: **String** name2 = "Michael";

String Methods

length() Returns the number of characters in this string. name1.length() => 4.

size() Returns the number of characters in this string. name2.size() => 7.

char charAt(int index); string subString(int startIndex, int endIndex); text1.concat(string text2)

Homework 8 / Lab 4

Consider the following Java Calculator application (partial code):

```
package Calculator;
import java.util.Scanner;

public class Calc {
	public static void main(String[] args) {
		int i=0, option=0;
		//create Scanner object inputValue with constructor Scanner() for user inputs
		Scanner inputValue = new Scanner(System.in);
		//double array of two dimensions to hold up to ten by two numbers from user
		double[][] arr_Two = new double[10][2];

		do {
			System.out.println("Enter the value for the option you want: ");
			System.out.println("1 for adding two values ");
			System.out.println("2 for displaying the operation(s) ");
			System.out.println("0 for quitting app. ");
			option = inputValue.nextInt();

			switch(option) {
			case 1 : {
				System.out.println("Enter the first value: ");
				arr_Two[i][0] = inputValue.nextDouble();
				System.out.println("Enter the second value: ");
				arr_Two[i][1] = inputValue.nextDouble();
				i++;
			}break;
			case 2 : {
				for(i=0; i < arr_Two.length; i++ ) {
					System.out.println(arr_Two[i][0] + " + " + arr_Two[i][1]
+ " = " + (arr_Two[i][0] + arr_Two[i][1]));
				}
			}break;
			case 0 : {
				int op = 0;

				System.out.println("Are you sure you want to quit? Enter 0 to
confirm, or other value to continue. ");
				op = inputValue.nextInt();
				if(op != 0) {
					System.out.println("You decide to continue. ");
					option = 4;
				}
				else {
					System.out.println("Bye. ");
					inputValue.close();
				}
			}break;
			default : {
				System.out.println("Enter 1,2 or 0 to quit! ");
			}break;
			}//end switch
		}while(option != 0);
	}//end main
}//end class Calc
```

This Java app. display a menu with 3 options for a simple calculator. For now, only the option 1 allow us to save the 2 numbers into an array of 2 dimensions and print out the full arithmetical expression. (10 + 20 = 30). The array can hold up to 10 additions of two numbers.
The second option is displaying the content of the array, the third one allow the user to quit the app. with confirmation.

Step 1.: Modify this code to add more options for operations like subtracting, multiplying and dividing and save the entered numbers similar like for adding. What are the problems, how you can solve them? Provide different options (at least one). Name and save this file Lab4ver1.java

Step 2.: Create methods (functions) in the class Calc to calculate and to return the result when call is made. Use the keyword static for all the methods.

```
public static double Add(double n1, double n2) {
      return n1 + n2;
}
```

Modify the call in the main() method in order to use a method that will return the result of the arithmetical operation once its call.

```
Add(arr_Two[i][0], arr_Two[i][1])
```

Test your all 4 methods, save and name this file Lab4ver2.java

Step 3.: Create another java class file under the same package, name it CalcMethods.java Place all 4 methods into this class, remove the **static** keyword. Create the object **c1** of this class inside the main() method:

```
//create object of CalcMethods class, calling the default constructor
            CalcMethods c1 = new CalcMethods();
```

Use this object to access the methods:

```
      c1.Add(arr_Two[i][0], arr_Two[i][1])
```

Test your application, save and name the file Lab4ver3.java

Identify yourself, add the date and a short description. Upload all .java files by LEA of Omnivox under the assignment: Lab 4 .

Thank you.