EEE 210: Software Engineering
Lab 3 Exercises for Week 4 (29 Jan. – 4 Feb.), Spring 2018

Note:

- Project folder nomenclature: Lab3_*yourname*
- After completion, zip your project folder and upload it to your Moodle account.
- Any queries during the lab should be discussed merely with the Instructor/TA.
- The output of each exercise you complete should be included (as screenshots) in the report named Report1_*yourname*. **Print and submit your report to the TA by the specified deadline**.

***Exercise 1*:** Write a Java program to convert a binary number to hexadecimal number.

Input A binary number
Output Corresponding hexadecimal and octal representation of the inputted number.

The output should look as follows:

```
Enter a binary number:  1101
Hexadecimal value:  D
Octal value:  15
```

***Exercise 2*:** Write a Java program to count the letters, spaces, numbers and other characters of an input string.

The output should look as follows:

```
Enter string: Aa kiu, I swd skieo 236587. GH kiu: sieo?? 25.33

No. of letters: 23
No. of spaces: 9
No. of numbers: 10
No. of other characters: 6
```

***Exercise 3*:** Write a Java method to display the middle character of a string. Note that:

(a) If the length of the string is odd, then there will be two middle characters.
(b) If the length of the string is even there will be one middle character.

A sample output is given below:

```
Input a string:  350asdqwe
The middle character in the string is s.
```

***Exercise 4*:** Write a Java program that takes a year from user and print whether that year is a leap year or not.

The output should look as follows:

```
Enter a year:  2016
2016 is a leap year.
```

Exercise 5: I have created an application that asks the user what they ate for various meals. Create a mealtime project directory in Eclipse with the following MealTime.java (file uploaded to Moodle as well) file in it:

```
import java.util.Scanner;

public class MealTime {

        public static void main(String[] args) {

                Scanner scan = new Scanner(System.in);

                System.out.println("What did you eat for breakfast?");
                String yourBreakfast = scan.next();
                System.out.println("You had " + yourBreakfast + " for breakfast.");

                System.out.println("\nWhat did you eat for lunch?");
                String yourLunch = scan.next();
                System.out.println("You had " + yourLunch + " for lunch.");

                System.out.println("\nWhat did you eat for dinner?");
                String yourDinner = scan.next();
                System.out.println("You had " + yourDinner + " for dinner.");

                scan.close();
  }
}
```

Even though we're asking the user about three separate meals, there's a lot of shared code. We need to DRY (**D**on't **R**epeat **Y**ourself) up the code.

(a) Define a method called `askWhatYouAteFor()` that will handle asking our user what they ate for various meals. Instead of repeating lines similar to `System.out.println("What did you eat for breakfast?");` three different times in our program, we'll simply call this method. Add this method to the MealTime class given above.

(b) Move everything related to each meal into *one* method call. For example, you can even move the `Scanner scan = new Scanner(System.in);` line into the `askWhatYouAteFor()` because it needs to be created inside the method that is going to use it.

Exercise 6: Consider the following car dealership Java program called App.java. Write the Vehicle class (name it Vehcile.java) for the above application. Your class should include all the instance variables in the above code, the constructor, and the method `worthBuying(int )`.

```java
import java.util.Scanner;

public class App {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        Vehicle hatchback = new Vehicle();
        hatchback.mYear = 1994;
        hatchback.mBrand = "Subaru";
        hatchback.mModel = "Legacy";
        hatchback.mMiles = 170000;
        hatchback.mPrice = 4000;

        Vehicle suv = new Vehicle();
        suv.mYear = 2002;
        suv.mBrand = "Ford";
        suv.mModel = "Explorer";
        suv.mMiles = 100000;
        suv.mPrice = 7000;

        Vehicle sedan = new Vehicle();
        sedan.mYear = 2015;
        sedan.mBrand = "Toyota";
        sedan.mModel = "Camry";
        sedan.mMiles = 50000;
        sedan.mPrice = 30000;

        Vehicle truck = new Vehicle();
        truck.mYear = 1999;
        truck.mBrand = "Ford";
        truck.mModel = "Ranger";
        truck.mMiles = 100000;
        truck.mPrice = 4000;

        Vehicle crossover = new Vehicle();
        crossover.mYear = 1998;
        crossover.mBrand = "Toyota";
        crossover.mModel = "Rav-4";
        crossover.mMiles = 200000;
        crossover.mPrice = 3500;

        Vehicle[] allVehicles = {hatchback, suv, sedan, truck, crossover};

        System.out.println("What is your maximum budget for a vehicle?");
        String stringUserMaxBudget = scan.next();
        int userMaxBudget = Integer.parseInt(stringUserMaxBudget);

        System.out.println("Here's what we have in your price range:");

        for ( Vehicle individualVehicle : allVehicles ) {
            if (individualVehicle.worthBuying(userMaxBudget)){
                System.out.println( "----------------------" );
                System.out.println( individualVehicle.mYear );
                System.out.println( individualVehicle.mBrand );
                System.out.println( individualVehicle.mModel );
                System.out.println( individualVehicle.mMiles );
                System.out.println( individualVehicle.mPrice );
            }
        }
        scan.close();
    }
}
```