Note all methods should be created with the static keyword

- For each question, write a method to solve the problem.
- Then call or invoke the method to ensure it works as required (expected). Testing should involve 100% code coverage.

Part 1- Method invocation

1. Create a method that when passed any given integer, will determine whether it is odd or even. It should output to the screen "**odd**" or "**even**"
2. Create a method that when passed any upper-case letter will convert it to a lower case letter. (Check first to be sure that it is not lower case already.)

```
public static void convertFromUpperToLower(char letter) {
        ….
}
```

Hint – use the Character class. E.g. **Character.isLowerCase()**

3. Create a method that when passed a pair of integers, will determine if the second is a multiple of the first.

```
public static void isMultiple(int num1, int num2){
        ….
}
```

4. Create a method that when invoked will generate a set of 10 random numbers between 1 and 100 (inclusive). Output should look similar to this…

```
Generating 10 random integers in range 1..100
Generated : 82
Generated : 52
Generated : 69
Generated : 59
Generated : 28
Generated : 28
Generated : 25
Generated : 65
Generated : 1
Generated : 13
Done
```

Hint you will need to inport the **import java.util.Random**;
Creating the class to use it : **Random rand = new Random();**

Check to Javaodc for the methods of the class.

http://docs.oracle.com/javase/6/docs/api/java/util/Random.html

5. Prompt the user for a number (between 5 and 50 (inclusive)). Then output to screen a countdown of the numbers starting with the given number and include on the same line the same number of asterisks.

E.g. 5 *****
4 ****
3 ***
etc….

Include validation – check if the number the user inputs is within range. If not, then output an error message.

Hint you will need a nested loop structure here to firstly print the number and then the required number of stars.

6. Prompt the user for a number. Write a method called **printNumberInWord** which prints "ONE", "TWO",... , "NINE", or "OTHER" if an **int** variable (passed as a parameter) is 1, 2,... , 9, or other, respectively.

   Firstly code this with a "**nested-if else**" statements; and then with a "**switch-case**" statement.

7. Create a method that will print the sum of 1+2+3, ..., to an upper bound determined by user input (e.g. 100). The program should also compute and display the average of these numbers. The output should look like:

   - The sum is 5050
   - The average is 50.5

   Hint : you may want to use a loop for this. The upper bound value should be passed as a parameter

8. Create a method that when invoked will determine if it is currently AM or PM. If AM then it should output "Good morning" and if PM then "I hope the morning went well for you. Enjoy the rest of your day."

   Hint : use *java.util.Calendar*

   To create a Calendar object (so that you can use the methods within)
   Calendar cal = Calendar.getInstance();

   The following field names can be used as an argument to the **Calendar.get(. . .)** method. In all of these examples, ***cal*** is a Calendar object.

Access Method	Meaning
cal.get(Calendar.YEAR)	int value of the year
cal.get(Calendar.MONTH)	int value of the month (0-11)
cal.get(Calendar.DAY_OF_MONTH)	int value of the day of the month (1-31)
cal.get(Calendar.DAY_OF_WEEK)	int value of the day of the week (0-6)
cal.get(Calendar.HOUR)	int value of the hour in 12 hour notation (0-12)
cal.get(Calendar.AM_PM)	returns either Calendar.AM or Calendar.PM

cal.get(Calendar.HOUR_OF_DAY)	int value of the hour of the day in 24-hour notation (0-24)
cal.get(Calendar.MINUTE)	int value of the minute in the hour (0-59)
cal.get(Calendar.SECOND)	int value of the second within the minute (0-59).
cal.get(Calendar.MILLISECOND)	int value of the milliseconds within a second (0-999).

There are several ways to do this.

Possible uses are **Calendar.HOUR_OF_DAY** and use if statement to compare answer to less than 12.

Or **Calendar.AM_PM** and compare value with 0 or 1 returned. Where 0 is AM and 1 is PM.

Part 2- Methods with return types

1. Create a method to calculate and return the area of a rectangle, given the *height* and *base*.
2. Given a number from 1-12, create a method to return the name of the appropriate month.
3. Create a method that takes **three int** arguments and **returns the average** of the numbers as a **double**. Note you may need to cast from an int to a double to get an accurate average result.
4. Create method that takes as an argument the name of a G8 country. It should then return the capital city of the country as a string. Remember to compare Strings you need to use **String.equals()**

Canada	Ottawa
France	Paris
Germany	Berlin
Italy	Rome
Japan	Tokyo
Russia	Moscow
United Kingdom	London
United States	Washington

5. Simulate tossing a coin. Prompt the user for a coin toss call i.e. heads or tails. Then simulate the coin toss and output if the user was right or wrong.

   public static void coinToss(String call)

   Hint to simulate a two outcome possibility e.g. 1 or a 2 you could use

   java.util.Random

   You could then map the outcome to head or tails. E.g. 1 = heads and 2 = tails.