# 44-542 Object Oriented Programming

# Lab01: Objects Lab Activity

**Objective:** Covers the usage of **String**, **Random**, and **Math** classes and its methods.

**NOTE:**

* For the **String** problems in this lab, use only **String** methods. You do not need to use arrays, split method, or any looping or selection constructs.
* Do not hard code any values unless specified and you must follow the given naming conventions.
* Check the given sample output to understand how the results need to be printed.
* Write comments appropriately for all the variables and methods.
* Generate **Javadoc** for all the classes of the project.
* Each class that you created for this assignment should include the information given below:

/\*\*

\* Class: 44542-NN Object-Oriented Programming

\* @author FirstName LastName

\* Description: Making sure everything works

\* Due: MM/DD/YY

\* I pledge that I have completed the programming assignment independently.

\* I have not copied the code from a student or any source.

\* I have not given my code to any other student and will not share this code with anyone under my circumstances.

\*/

Where NN is the section number.

1. Create a New Project in NetBeans and name it as **Lastname\_Lab01Objects** where **Lastname** is your last name.
2. Create a new package in the above project created and name it as **objects**.
3. Create a new Java Main Class in the above package and name it as **StringsAndNumbers.** The@author annotation must contain your full name.
4. Use the same **StringsAndNumbers** class to answer all the below questions.

**Questions**

**// String Class**

1. Inside the Main method do the following:

String greet = **“Hello Bearcats welcome to spring 2024 Object Oriented Programing class”;**

* + 1. print the value of greet
    2. print the length of the value of greet
    3. Print the value of greet but place **Object Oriented Programing** inside a two double quotes (see sample output).

1. Print **Object Oriented Programing** using greet and **substring ()** function.
2. Check whether the string stored in greet starts with **Hello** or not?
3. Check whether the string stored in greet ends with **Programing** or not?
4. Create 3 Strings **“My name is LastName\_FirstName”, “My favorite programming language is LanguageName because”,** and **“Reason to like(Give your own reason)”.** Store them into variables intro,favSubject and reason.
5. Concatenate three strings intro, favSubject and reason with a Full Stop(**.**) between them. Store the concatenated string into a new variable about. (Do not use ‘+’ operator use join () method instead)
6. Print the position of the second occurrence of **“.”** (Full Stop) in the String about.
7. Replace the second occurrence of **“.”** with a white space and print the string after replacing.
8. Extract your favorite language name from the string about and store it into a new variable language. (Use **Substring()** method)
9. Repeat the String stored in language 5 times without using loops.
10. Create a new string **“ This is my first Java Lab Activity ”** and store them into a string variable firstLab.
    * 1. Convert all the letters stored in firstLab into lower case and store them into another string variable lowerLab.
      2. Use **equals ()** method to check the equality between firstLab and lowerLab andprint the result.
      3. Use **equalsIgnoreCase()** method to check the equality between firstLab and lowerLab andprint the result.
      4. Replace all the occurrences of character **‘i’** in firstLab with a character **‘x’** and print**.**
      5. Trim the whitespace in firstLab and print the string after trimmed.

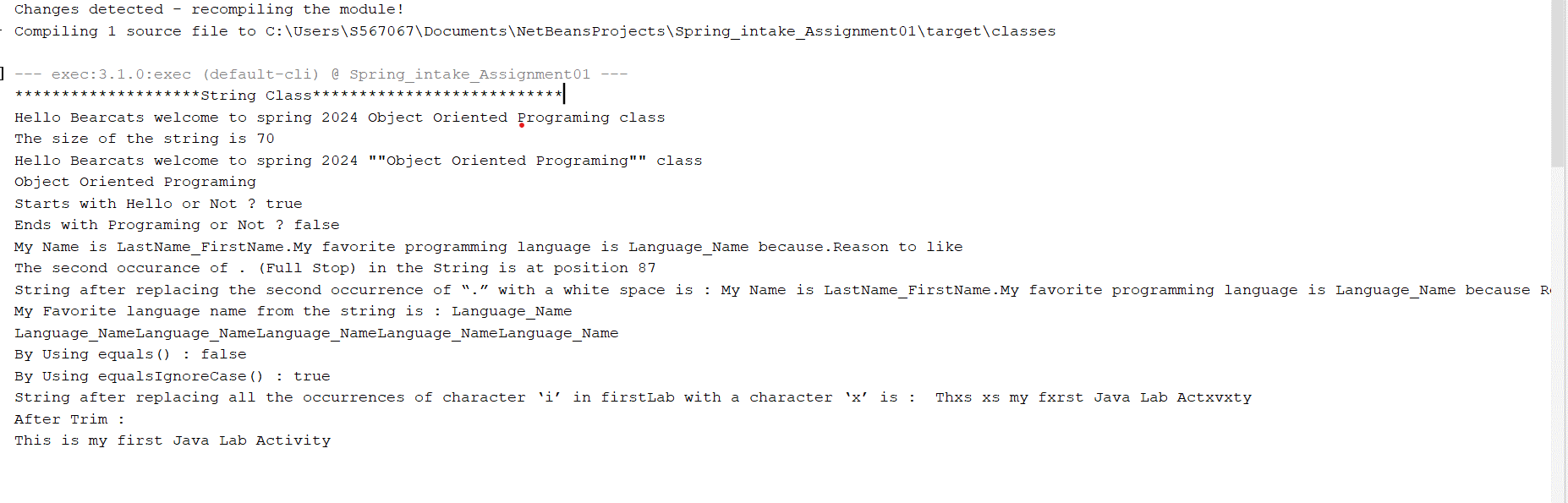
**//Math Class**

1. Methods in the **Math** class are all static methods. Static methods do not require an instance of the class to be invoked. Instead, you use the class name, followed by a dot, followed by the method. Use the Math class to solve the following problems. Please refer to the Math class API for more information.
   1. Write statements below:
      1. Evaluate the expression *√* (93+243) and print the rounded value of the result.
      2. Using the Math class, print the value of and . Compare these values and print whether you get the same results or not.
      3. Print the **ceil** value of the expression .
      4. Print the value of log95 and log32.
   2. Consider a line, where m is slope and b is intercept, passing through two points, namely (x1, y1) = (1, 4) and (x2, y2) =(3, 8). Compute the slope and intercept of the line and print the equation of the line.

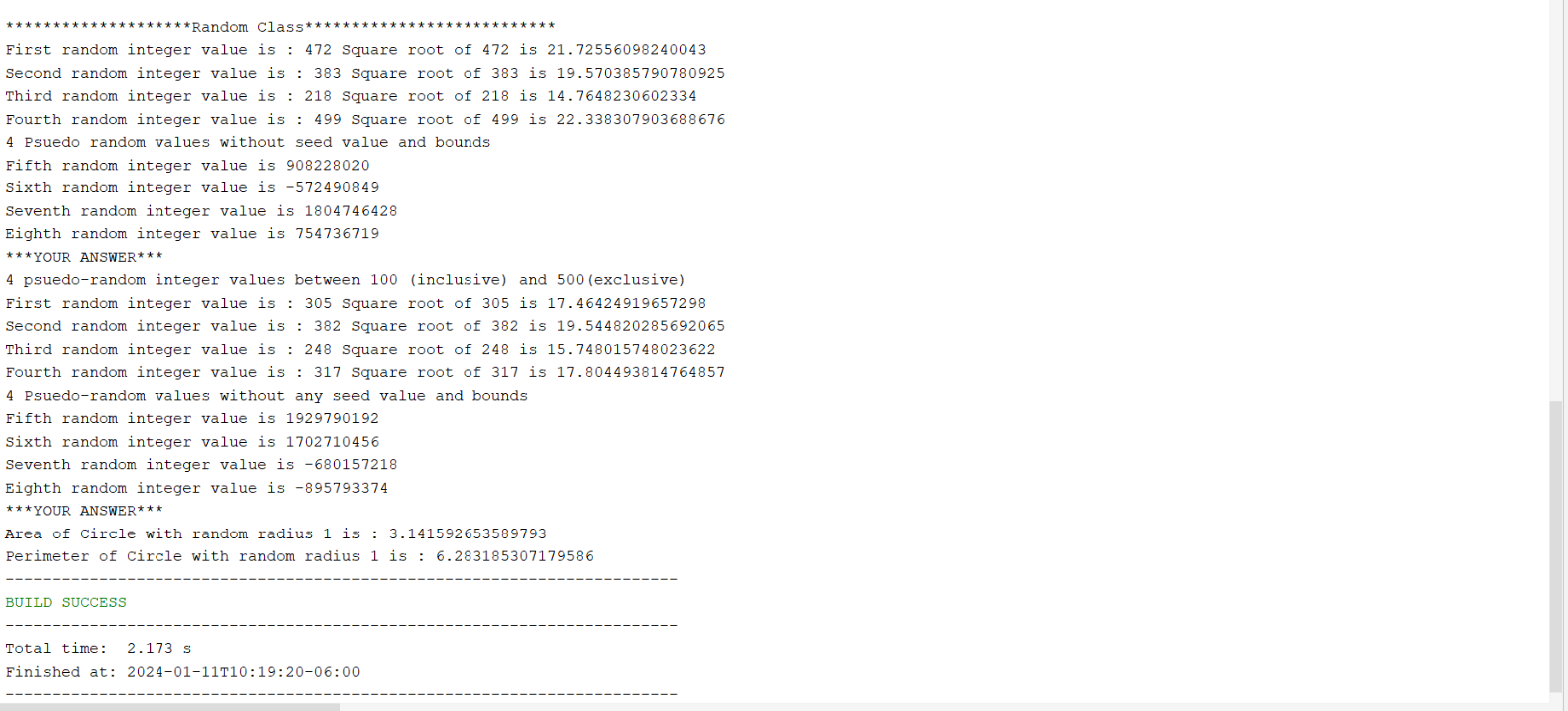
**// Random Class**

1. The **Random** class can be used to generate pseudorandom numbers – they look like random numbers, and they act like random numbers, but they aren’t quite random. For help in completing the following problems, refer to **(Random class Java API)**
   1. Create an instance of the **Random** class using *no* seed value. Generate and print 4 pseudo-random integer values between 100 (inclusive) and 500 (exclusive) and their square root.
   2. Generate 4 random values without passing the seed and bound values and print them.
   3. Run your program 3 to 4 times. Did you get same values every time? Write your answer in print statement.
   4. Create an instance of the **Random** class using with seed value 40. Generate and print 4 pseudo-random integer values between 100 (inclusive) and 500 (exclusive) and their square root.
   5. Generate 4 random values without passing the seed and bound values and print them.
   6. Run your program 3 to 4 times. Did you get same values every time? Write your answer in print statement with a valid explanation.
2. Create another instance of the **Random** class name it as random2. Generate a random integer value between 1 and 10 and store it in a variable randomRadius. Using Math class methods (Use Math.PI and Math.pow ()) calculate the **area** and **perimeter** of the circle and print .

**Sample Output:** The output should be printed as it is in the below box.







**Submit your solution by following the steps below:**

* Save your files in NetBeans.
* Use **Shift+Alt+F** to format your code.
* Zip your entire Project. (File name should be *Lastname*\_Lab01Objects.zip where Lastname is your last name.)
* Submit the Zip file to the Lab01Objects Dropbox.
* Download the Zip file you have submitted.
* Look in the Zip file and verify that **StringsAndNumbers.java** is correct. If not resave your project in NetBeans and resubmit.