**Project 3: Email Generator**

In this Java programming assignment, you will practice using the String type and the built-in String functions in order to create functions to be used to generate e-mail addresses.

1. Design a class called **EmailGenerator** in a file called **EmailGenerator.java**. This class will hold the main method, and the class methods that we will write in this assignment.
2. Write an empty **public static void main(String[] args)** method. This method should appear inside the curly braces of the EmailGenerator class.
3. Write two static class methods (**makeUserName, makeEmail**). The method **makeUserName** should take two String arguments and will return a String. The method should combine the first letter of the first String and combine it with the full second String, returning the final String in all lowercase characters. The method **makeEmail** should take two String arguments and will return a String. This method should combine the first string, a “@” symbol, and the second String. Several example method calls appear below.

**EmailGenerator.makeUserName(“Alan”,”Turing”); // returns “aturing”**

**EmailGenerator.makeEmail(“aturing”,”purdue.edu”); // returns “aturing@purdue.edu”**

1. Complete the definition for main. Your program should create a new **Scanner** object, prompt the user to type in a first and last name, and then collect two Strings. The **main** method should then call the two static methods of the **EmailGenerator** object, first creating a username String and then generating the corresponding email String. You can pick any domain for your email (we used “purdue.edu” above). The program should output the final email address for a user with the given first and last name. For example, if the user enters “Grace” and “Hopper” as inputs:

“This user’s e-mail is: ghopper@purdue.edu”

1. Submit your **EmailGenerator.java** file on Blackboard.

For those students that feel confident and would like an extra challenge, add the following elements to your code before submitting.

1. Make a minor adjustment to your **makeUserName** method, such that the total length of the username is exactly 8 total characters. If the last name is more than 7 characters, the first seven characters of the last name should be appended to the first character of the first name. If the last name is seven characters or fewer, the generated username should include as many characters from the first name that will make up the difference. If both first and last name are less than 8 letters combined, add enough ‘x’ characters to make the username String a total of 8 characters.