Recitation 7 Practice 2

1. Make a directory named **r7** in your **csci101** directory.
2. In your **csci101/r7** directory, create a program in a file named **Practice2.java** that satisfies the Program Requirements shown below.

Program Requirements

1. Print to the screen the string **Recitation 7 Practice 2.**
2. Print to the screen the string **----------**.

3. Write a method named **printName** that simply prints your named to the screen.
   Call **printName** in your **main** method.

4. Write a method named **printLocation** that has two String parameters, the first named **city** and the second named **state**. The method prints to the screen the value held in the parameter **city** followed by a comma and then the value in the parameter named **state**.
5. Call **printLocation** in **main**, passing to it the strings **Bridgewater** and **VA**.

6. Write a method named **getTen** that returns the value **10**.
7. Invoke **getTen** in the **main** method and store the return value in a variable named **ten**. Print **Ten:** followed by the value in the variable **ten**.

8. Write a method named **tripleIt** that takes an integer as an argument and returns **3** times the value passed to it.
9. Call **tripleIt** in **main**, passing to it the value **5**. Store the return value in a variable named **result**. Print **Triple:** followed by the value in the variable named **result**.

10. Write a method named **max** that takes 2 integers as arguments and returns the larger of the two values passed to the method**.**
11. Invoke the **max** method in **main**, passing to it the values **20** and **2** and store the value returned by **max** in a variable named **largest**. Print **Largest:** followed by the value held in the variable **largest**.

12. Declare a method named **createArray** that has one integer parameter and returns an array of integers. The method allocates an array of integers whose length is equal to the value passed to the method.
13. In your main method declare an integer array named **arr1** and initialize it with the reference returned by **createArray.**

14. Declare a method named **printArray** that takes an array of integers as an argument. When invoked, the method prints to the screen, on a single line, the elements in the array passed to, with spaces between the elements.
15. Call **printArray** in main, passing to it **arr1**.

16. Declare a method named **setArray** that has two parameters, the first being a reference to an array of integers, and the second being an integer. When invoked, the method sets all of the elements in the array reference by the first parameter equal to the value in the second parameter.
17. Call **setArray** in main, passing to it **arr1** and **-1** and then call **printArray** in main, passing to it **arr1**.