CSCI-101 Exam 2

Name _____

Instructions:

Please follow the rules below as you work through this exam.

- Please leave all notebooks and electronics (including cell phones and smart watches) at the side of the room.
- This is a closed book/closed notes exam.
- **Do not spend too much time on any one problem.** You have 50 minutes to complete this exam.
- Partial credit is awarded.
- Please write legibly. If I cannot read your answers, I cannot give you credit.
- Please write your answers **in the order specified**. If you need additional paper, please raise your hand to ask your instructor for additional paper.
- Your code must be written to behave as specified.
- You must properly use all identifiers that are explicitly stated.
- Please use proper and consistent coding conventions (spacing, naming identifiers, etc.).
- Please stay in your seat until you are ready to hand in your exam. You may leave when you are finished.
- Once you leave the testing room you cannot return until the exam is over. If you need to use the restroom, please use it now.

Instructions for Remote Students

- You are not allowed to use any resources *including the compiler*. You will receive a score of 0 if you are found to be using your compiler.

- Keep your eyes on your computer screen for the duration of the exam. If the instructor suspects you are using resources that are not allowed you will have to retake the exam as an oral exam with the instructor.

- Share your entire screen.

- Open your terminal application and place the terminal application in one half of the screen. Place this document in the other half of the screen.

- In the terminal application, log into cs.bridgewater.edu.

- **In your home directory**, create a directory named **exam2** and change your working directory to **exam2**.

- Create a file named **Exam2.java.** Write a complete program that satisfies the program requirements shown below.

- When you are finished, let me know in the Chat pane. I will then copy your exam files to my computer.

1. In main, print to the screen the string of characters **Exam 2.**
2. In main, print to the screen the string of characters **_____**.

3. In main, write a statement that creates a Scanner that can be used to read data from the keyboard.

4. In main, ask the user to input a String containing the name of a musician.
5. In main, read the value typed on the keyboard into a variable named **musician**.
6. In main, print to the screen **Number of characters:** followed by the number of characters in the string referenced in the variable named **musician.**

7. In main, declare an array named **arr1** that contains the values **5**, **6**, and **7**.

8. In main, declare an array named **arr2** that can hold 10 integers.
9. In main, ask the user to enter 10 integers. Read the 10 integers from the keyboard and store them in the array referenced by the variable named **arr2**.

10. Create a method named **printArray** that takes an array of integers as an argument and prints the values in the array to the screen on a single line.

11. In main, print the values in **arr2** by calling **printArray**.

12. Declare a method named **contains** that has an array of integers as the first parameter and an integer as the second parameter. The method returns **true** if the integer that is passed in as the second argument is contained in the array that is passed as the first argument; otherwise the method returns **false**.

13. In main, invoke **contains** while passing to it **arr1** and the value **8**. Print to the screen **Contains 8:** followed by the value returned by **contains**.

14. Declare a method named **isSquare** that takes a 2D array of integers as an argument. The method returns **true** if all of the rows in the 2D array have the same number of elements and the number of rows in the 2D array equals the number of columns in the 2D array; otherwise the method returns **false**.

15. In main, declare a 4x3 2D array of integers named **matrix** that contains the following values:
    row 1: **1, 2, 3**
    row 2: **4, 5, 6, 7**
    row 3: **8, 9**
16. In main, invoke **isSquare**, passing to it the 2D array referenced in **matrix**. Print to the screen **Is square:** followed by the value returned by **isSquare**.