

# CSCI-101 Programming 1

## Lab 12

### INSTRUCTIONS

Create a directory named **lab12** in your **labs** directory. Inside the **lab12** directory create a class named **Lab12** and inside **Lab12**'s **main()** method write the necessary code to satisfies the following:

- ☐ Declare a **Scanner** that can read from the keyboard.
- ☐ Ask the user to enter an integer, then read the integer from the keyboard and store the value in a variable named **input1**.
- ☐ Increment the value in **input1**.
- ☐ Print to the screen "**input:** " followed by the value in the variable named **input1**.
- ☐ Use an if-else block to print "**odd**" if the value in **input1** is odd or print "**even**" if the value is even.
- ☐ Ask the user to enter their name.
- ☐ Read a String from the keyboard and store the value in a variable named **name**.
- ☐ Use the **?:** operator to set a variable named **bigName** to **true** if the string contains more than 15 characters or sets **bigName** to **false** if it does not.
- ☐ Print to the screen "**bigName:** " followed by the value stored in the variable named **bigName**.
- ☐ Ask the user to enter an integer between **1** and **3**.
- ☐ Read the value from the keyboard and store it in a variable named **input2**.
- ☐ Use a switch statement to print to the screen the english word for the value in **input2** if the value is between 1 and 3; otherwise print to the screen "**some other value**".
- ☐ Use a do-while loop to print the numbers from **3** to **37**, inclusively.
- ☐ Use a while-loop to print the numbers from **3** to **37**, inclusively, from largest to smallest.
- ☐ Use a for-loop to compute the sum of the numbers between **3** to **27** (inclusively). Store the sum in a variable named **sum**.
- ☐ Print to the screen "**Sum:** " followed by the value in the variable named **sum**.
- ☐ Create an array named **array1** that can hold 20 integers.

- ☐ Use a for-loop to populate **array1** with 20 random integers between **1** and **10** (inclusively).
- ☐ Use a for-loop to print the values in **array1** to the screen on a single line.
- ☐ Declare a variable named **val1** and initialize it to the last element in **array1**.
- ☐ Print to the screen "**val1:** " followed by the value in **val1**.
- ☐ Use a for-loop to add 1 to all of the elements in **array1**.
- ☐ Use a for-loop to print the values in **array1** to the screen on a single line.
- ☐ Declare a **10 x 10** array of integers named **matrix1**.
- ☐ Use nested for-loops to initialize all of elements in **matrix1** to random values between **0** and **1** (inclusively).
- ☐ Use nested for-loops to print to the screen all of the elements in **matrix1**, with the elements in each row on a single line, and each row on separate lines.
- ☐ Use nested for-loops to set all of the elements in the 5th and 6th rows of **matrix1** to the value **2**.
- ☐ Use nested for-loops to print to the screen all of the elements in **matrix1**, with the elements in each row on a single line, and each row on separate lines.
- ☐ Declare a variable named **val2** and initialize it to the value at row 3, column 3 in **matrix1**.
- ☐ Print to the screen "**val2:** " followed by the value in the variable named **val2**.
- ☐ Print to the screen "**Sum:** " followed by the sum of all of the elements in **matrix1**.
- ☐ Print to the screen "**Num odd:** " followed by the number of elements in **matrix1** that are odd.
- ☐ Outside of **main()** write a method named **max** that has two integer parameters and returns the larger of the two values that are passed into the method.
- ☐ Inside **main()** call **max** passing to it **val1** and **val2**. Store the value that is returned by **max** in a variable named **largest**.
- ☐ Outside of **main()** write a method named **getLargest** that takes an array of integers as an argument and returns the largest value in the array.
- ☐ Inside **main()** call **getLargest**, passing to it **array1**. Store the value returned by **getLargest** in a variable named **largest**. Print to the screen "**largest:** " followed by the value in the variable named **largest**.

- ☐ Outside of **main()** write a method named **countOccurrences** that has two parameters. The first is an array of integers named **arr** and the second is an integer named **k**. The method returns the number of integers in **arr** that have the value **k**.
- ☐ Inside **main()** call **countOccurrences**, passing to the method **array1** and the value **2**. Store the value that is returned by the method in a variable named **count**. Print to the screen "**count:** " followed by the value in the variable named **count**.
- ☐ Outside of **main()** write a method named **printMatrix** that takes a 2D array of integers as an argument and prints the values in each row of the matrix on a single line, with each row on a separate line.
- ☐ Inside **main()**, call **printMatrix** passing to it **matrix1**.
- ☐ Outside of **main()** write a method named **setRandom** that takes a 2D array of integers as an argument and randomly sets all of the elements in the 2D array to either **0** or **1**.
- ☐ Inside **main()**, use an infinite-loop to repeatedly call **setRandom** and **printMatrix** (passing to them **matrix1**) until the matrix that is printed (when squinting at it) resembles a monkey at a keyboard printing the complete works of Shakespeare.