

## CMSC 131 Quiz 2 Worksheet

The next quiz for the course will be on Mon, Feb 25. The following list provides additional information about the quiz.

- The quiz will be a written quiz (no computer).
- Closed book, closed notes quiz.
- The quiz will be in lab session.
- Answers must be neat and legible.
- Quiz instructions can be found at <http://www.cs.umd.edu/~nelson/classes/utilities/examRules.html>.
- Regarding Piazza - Feel free to post questions in Piazza regarding the worksheet and possible solutions to problems.
- **You must take your quiz in your assigned lab/discussion session and not show up to a random discussion session. We will not grade quizzes taken in the incorrect session.**
- We use the Gradescope system to grade your quizzes after they have been scanned. For the system to recognize your work, you **need to print your name (uppercase) and student id**. The following is an example of the information you need to provide in your quiz:

FIRSTNAME, LASTNAME (PRINT IN UPPERCASE):

MARY, SMITH

STUDENT ID (e.g. 123456789):

123456789

**The following exercises cover the material to be included in this quiz.** Solutions to these exercises will not be provided, but you are welcome to discuss your solutions with the TAs or instructor during office hours. It is recommended that you try these exercises on paper first (without using the computer). For input and output use the Scanner class. For this quiz you are not responsible for methods associated with the JOptionPane class (e.g., showInputDialog, showMessageDialog).

### Exercises

1. When do we use a do while?
2. When do we use a for loop?
3. What is short-circuiting?
4. Write a program that reads two integer values and prints the even numbers in that range.
5. Write a complete program that computes the factorial of a number. For example, factorial of 4 (4!) is 24.
6. Write a complete program that prints the sum of odd numbers between min (inclusive) and max (inclusive). The program will read the values min and max and display the sum. You can assume min and max are integer values.
7. Write a complete program that reads two integer values representing the number of rows and columns of a rectangular diagram. After reading the values, the program will draw the rectangular diagram using the \* character. For example, if the user enters 2 and 3, your program will print:

```
***  
***
```

**You must use a while loop.**

8. Write a complete program that reads an integer value representing the number of rows associated with a triangular diagram. After reading the value, your program will generate a triangular diagram with the specified number of rows. For example, if the user enters 4 your program will print:

```
*  
**  
***  
****
```

**You must use a for loop.**

9. Modify the previous program so the program also reads the character to use in the diagram.
10. Modify the previous program so the program reads two characters; the first one will be used for odd-numbered rows and the second for even-numbered rows.
11. Write a complete program that reads an integer value representing the number of rows associated with a triangular diagram. After reading the value, your program will generate a triangular diagram with the specified number of rows. For example, if the user enters 4 your program will print:

```

*
**
***
****

```

**You must use a for loop.**

12. Write a complete program that reads an integer value (let's called it n). The program will then generate a diagram with  $2n - 1$  rows and with a triangular shape as illustrated in the examples below.

Diagram for a value of  $n = 3$

```

*
**
***
**
*

```

Diagram for a value of  $n = 4$

```

*
**
***
****
***
**
*

```

13. Write a program that reads an integer and prints the number in binary. For example, if the user enters 8 the program will display 1000. To compute the binary number, divide the value provided by the user by 2 until you get 0. The set of remainders that are generated (when read from right to left) represent the binary number. **Hint:** Notice that the result does not need to be a number; it is just a string with the binary number that corresponds to the value provided by the user.
14. Write a program that prints a triangle. The program will read the size (an integer value) of the triangle and a character. It will then generate a triangle with a number of rows that corresponds to size and where the character provided is used for even-numbered rows. Odd-numbered rows will always use a \*. Use the message "Enter size:" and "Enter character:" to read data. Below we have provided two examples of running the program. Notice your program must work for other values. Underlined text represent input provided by the user.

```

Enter size: 4
Enter character: #
****
###
**
#

```

```

Enter size: 6
Enter character: #
*****
#####
****
###
**
#

```