

Assignment #1

CSC111: Introduction to Programming

Learning Outcomes

When you have completed this assignment, you should understand:

- How to design, run and check a simple and complete a C program on your own.
- The effect of escape sequences on printed strings.
- How to write basic functions.
- How to create and assign values to variables.
- How to use format specifiers to format output.
- The flow of control using functions (the effects of function calls and assignment statements)
- How to format and document a C program.

Assignment Overview

1. Create a new file called `assignment1.c` in Jupyter Hub
2. Design the functions according to the specifications in the Function Specifications section.
3. Test to make sure your code produces the expected output.
4. Ensure that the filename is `assignment1.c` (files with an incorrect name will not be marked).
5. Download your `assignment1.c` file from Jupyter Hub (File→Download) and submit this file to the Assignment 1 dropbox (under Assignments) on the CSC111 BrightSpace Page.
6. Download your file from BrightSpace to confirm that name and file contents are as expected.

Reminder: Your code is to be designed and written by only you and not to be shared with anyone else. See the Course Outline for details explaining the policies on Academic Integrity. Submissions that violate the Academic Integrity policy will be forwarded directly to the Computer Science Academic Integrity Committee.

Grading

1. Late submissions will not be accepted
2. You must submit a file with the name `assignment1.c` or you will be given a **zero grade**.
3. Your function names must match the specification **exactly** or you will be given a **zero grade**.
4. None of the functions you write in this assignment should take arguments - you will be given a **zero grade** for any functions that take arguments.
5. Any code written in the main function may not be marked.
6. We will do spot-check grading for code quality. All assignments are graded BUT only a subset of your code might be graded for code quality. You will not know which portions of the code will be graded, so all of your code must be complete and adhere to specifications to receive marks.
7. Your code must run without errors in the Computer Science department's Jupyter Hub environment using the compilation command provided in this document.
8. We reserve the right to replace your main function with our own, allowing it to better interact with our testing software.

Marks will be given for:

- Your code producing the correct output.
- Your code following good coding conventions
 - Proper indentation
 - Documentation (comments)
 - Use of whitespace to improve readability
 - Names of variables should have meaning relevant to what they store

This section will describe the functions that you are tasked to create. When testing your code type the following command to compile your code:

and the following to run the code:

Printing Quotes

1. Design a function named `quote()`, that meets the following criteria:

- "I'm not sleeping, I'm inspecting the inside of my eyelids" - Hawkeye Pierce

- ```
Terminal 1
jovyan@jupyter-jkrysl:~$ gcc -Wall -Werror -pedantic -std=c18 -o assignment1 assignment1.c
jovyan@jupyter-jkrysl:~$./assignment1
"I'm not sleeping, I'm inspecting the inside of my eyelids" - Hawkeye Pierce
jovyan@jupyter-jkrysl:~$
```

2. Design a function, named `framed_quote()`, that meets the following criteria:

- ```

/*****\
|"I'm not sleeping, I'm inspecting the inside of my eyelids" - Hawkeye Pierce|
/*****\

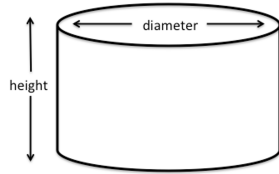
```

```
jovyan@jupyter-jkrysl:~$ gcc -Wall -Werror -pedantic -std=c18 -o assignment1 assignment1.c
jovyan@jupyter-jkrysl:~$ ./assignment1
/*~~~~~**~~~~~**~~~~~**~~~~~**~~~~~**~~~~~**~~~~~**~~~~~**~~~~~**~~~~~**~~~~~*\
|*"I'm not sleeping, I'm inspecting the inside of my eyelids" - Hawkeye Pierce*|
\*****~*****~*****~*****~*****~*****~*****~*****~*****~*****/
jovyan@jupyter-jkrysl:~$
```

2

Math Calculations

Design a function `print_surface_area()` that will calculate the surface area of a cylinder.



Given the height of the cylinder is 1.11m and the diameter is 6m (see diagram). Ensure that you adhere to the function specifications below:

1. Implement the `print_surface_area()` function as follows:
 - Create 2 variables: one for height and set it to 1.11 and another for the diameter, and set it to 6.
 - Using the diameter and the value of π , write a statement to calculate the circumference of the cylinder. Store the result of the calculation
 - For the value of π it is suggested that you use a symbolic constant. **DO NOT** use the constant in the C's math library (if you are not familiar with this, don't worry - please define your own value for π).
 - Using the diameter and the value of π , write a statement to calculate the area of the top of the cylinder and store the result.
 - Using the height and the circumference, that you calculated earlier, write a statement to calculate the area of the walls of the cylinder.
 - Using the two variables created above that are storing the area of the top and the area of the walls, calculate the total surface area of the cylinder.
 - Print the resulting area to the screen with a two (2) decimal place precision and the units, as shown in the sample output.
2. Edit and run your program until your output is as shown below. The accuracy of your calculation can vary as it will be dependent on the value you used for π . In the test run for the output below, we defined π as 3.1415.

```
jovyan@jupyter-jkrysl:~$ gcc -Wall -Werror -pedantic -std=c18 -o assignment1 assignment1.c
jovyan@jupyter-jkrysl:~$ ./assignment1
The total surface area is 77.47 meters squared
jovyan@jupyter-jkrysl:~$
```

Figure 3: The terminal window after compiling and running the `print_surface_area()` function

NOTE: There are other algorithms to calculate the surface area of a cylinder but we are asking you to follow this algorithm to give you practice with the creation and use of variables.