

# CS 161A: Programming and Problem Solving I

## Assignment A03 Sample Algorithmic Design Document

---

*Make a copy before you begin (File -> Make a copy). Add the Assignment # above and complete the sections below BEFORE you begin to code. The sections will expand as you type. When you are finished, download this document as a PDF (File -> Download -> PDF) and submit to D2L.*

*This document contains an interactive checklist. To mark an item as complete, click on the box (the entire list will be highlighted), then right click (the clicked box will only be highlighted), and choose the checkmark.*

Planning your program before you start coding is part of the development process. In this document you will:

- ☒ ~~Paste a screenshot of your zyBooks Challenge and Participation %~~
- ☒ ~~Paste a screenshot of your assigned zyLabs completion~~
- ☒ ~~Write a detailed description of your program, at least two complete sentences~~
- ☒ ~~If applicable, design a sample run with test input and output~~
- ☒ ~~Identify the program inputs and their data types~~
- ☒ ~~Identify the program outputs and their data types~~
- ☒ ~~Identify any calculations or formulas needed~~
- ☒ ~~Write the algorithmic steps as pseudocode or a flowchart~~
- ☐ Tools for flowchart - [Draw.io](#) - [Diagrams.net](#)

### 1. zyBooks

---

Add your zyBooks screenshots for the % and assigned zyLabs completions below. Required percentages: all **assigned** zyLabs, Challenge Activity with at least 70%, and Participation Activity with at least 80%.

**Challenge and Participation % screenshot:**

3. CS 161A: Data Types & Math Functions		100%	100%	100%	^
3.1 Using math functions		100%	100%		v
3.2 Integer division and modulo		100%	100%		v
3.3 Type conversions		100%	100%		v
3.4 Binary			100%		v
3.5 Characters		100%	100%		v
3.6 Strings		100%	100%		v
3.7 Input stream and cin functions				No activities	
3.8 Input Failure				No activities	
3.9 Integer overflow			100%		v

Assigned zyLabs completion screenshot:	
3.17 LAB: Phone number breakdown	100% ^
<div> <div></div> <div>Lab activities</div> </div> <div>User score: 10 / 10 points</div>	
3.18 LAB: Input: Mad Lib	100% ^
<div> <div></div> <div>Lab activities</div> </div> <div>User score: 10 / 10 points</div>	

## 2. Program Description

In the box below, describe the purpose of the program. You must include a detailed description with at least two complete sentences.

### Program description:

This program is for Oreo Cookie's website and will calculate user's servings (double) and calories (double) based on user input of the whole amount of cookies (integers). There is a strong emphasis on wanting the outputs to have .0 singular, trailing decimal points even for whole numbers.

## 3. Sample Run

If you are designing your own program, you will start with a sample run. Imagine a user is running your program - what will they see? What inputs do you expect, and what will be the outputs from the given inputs? Choose test data you will use to test your program. Calculate and show the expected outputs. Use the sample run to test your program.

### Sample run:

```
Welcome to Oreo Cookie's Calorie and Serving Calculator!

Enter the number of Oreo Cookies using a whole number: 12

With 12.0 cookies, you will have 2.4 servings and 384.0 calories.
Happy snacking!
-----
Oreo Cookie Amount: 12.0
Oreo Cookie Servings: 2.4
Oreo Cookie Calories: 384.0
-----
Thank you for using the Oreo Cookie Calorie and Serving Calculator
program!
```

## 4. Algorithmic Design

Before you begin coding, **you must first plan out the logic** and think about what data you will use to test your program for correctness. All programmers plan before coding - this saves a lot of time and frustration! Use the steps below to identify the inputs and outputs, calculations, and steps needed to solve the problem.

### Algorithmic design:

a. Identify and list all of the user input and their data types.

- o numCookies = 0 (integer)

1. Will be converting this to a <b>double</b> using <code>static_cast&lt;double&gt;()</code> for outputs
b. Identify and list all of the user output and their data types.
<ul style="list-style-type: none"> <li>○ <b>numServings</b> (<b>double</b>)</li> <li>○ <b>numCalories</b> (<b>double</b>)</li> <li>○ <b>numCookies</b> (<b>integer</b> converted to <b>double</b>)</li> </ul>
c. What calculations do you need to do to transform inputs into outputs? List all formulas needed, if applicable. If there are no calculations needed, state there are no calculations for this algorithm.
<ul style="list-style-type: none"> <li>○ <b>SET COOKIES_PER_SERVING</b> as <b>integer constant</b> to 5 (cookies per serving)</li> <li>○ <b>SET CALORIES_PER_SERVING</b> as <b>integer constant</b> to 160 (servings per calorie)</li> <li>○ <b>numServings</b> (<b>double</b>) = <b>numCookies</b> (converted to <b>double</b>) / <b>COOKIES_PER_SERVING</b> (<b>constant integer</b>) <ul style="list-style-type: none"> <li>1. I.e. <b>numServings</b> = <code>static_cast&lt;integer&gt;(numCookies) / COOKIES_PER_SERVING;</code></li> </ul> </li> <li>○ <b>numCalories</b> (<b>double</b>) = <b>numServings</b> * <b>CALORIES_PER_SERVING</b>;</li> </ul>
d. Design the logic of your program using pseudocode or flowcharts. Here is where you would use conditionals, loops or functions (if applicable) and list the steps in transforming inputs into outputs. Walk through your logic steps with the test data from the assignment document or the sample run above.
<ol style="list-style-type: none"> <li>1. <b>SET COOKIES_PER_SERVING</b> as <b>integer constant</b> = 5</li> <li>2. <b>SET CALORIES_PER_SERVING</b> as <b>integer constant</b> = 160</li> <li>3. <b>DECLARE</b> variable <b>numCookies</b> as <b>integer</b> = 0</li> <li>4. <b>DECLARE</b> variable <b>numCalories</b> as <b>double</b> = 0</li> <li>5. <b>DECLARE</b> variable <b>numServings</b> as <b>double</b> = 0</li> <li>6. <b>DISPLAY</b> welcome message as <b>string literal</b></li> <li>7. <b>DISPLAY</b> prompt for number of cookies as <b>string literal</b></li> <li>8. <b>INPUT</b> into <b>numCookies</b></li> <li>9. <b>SET numServings</b> = ((<b>numCookies</b> converted to <b>double</b>) / <b>COOKIES_PER_SERVING</b>)</li> <li>10. <b>SET numCalories</b> = (<b>numServings</b> * <b>CALORIES_PER_SERVING</b>)</li> <li>11. <b>DISPLAY</b> "With " <b>numCookies</b> " cookies, you will have " <b>numServings</b> " servings and " <b>numCalories</b> " calories. Happy snacking!" <b>string literal</b> with <b>variables displayed and set to 1 digit precision</b>.</li> </ol>

12. **DISPLAY** "-" 25 times to form a line, align-ed **left** using **setfill**
13. **DISPLAY** Oreo Cookie Amount **string literal** + **numCookies** set to 1 digit precision
14. **DISPLAY** Oreo Cookie Servings **string literal** + **numServings** set to 1 digit precision
15. **DISPLAY** Oreo Cookie Calories **string literal** + **numCalories** set to 1 digit precision
16. **DISPLAY** "-" 25 times to form a line, align-ed **left** using **setfill**
17. **DISPLAY** thank you message as **string literal**
18. **END** program

## 5. Pseudocode Syntax

Think about each step in your algorithm as an action and use the verbs below:

To do this:	Use this verb:	Example:
Create a variable	DECLARE	DECLARE integer num_dogs
Print to the console window	DISPLAY	DISPLAY "Hello!"
Read input from the user into a variable	INPUT	INPUT num_dogs
Update the contents of a variable	SET	SET num_dogs = num_dogs + 1
<b>Conditionals</b>		
Use a single alternative conditional	IF <i>condition</i> THEN <i>statement</i> <i>statement</i> END IF	IF num_dogs > 10 THEN DISPLAY "That is a lot of dogs!" END IF
Use a dual alternative conditional	IF <i>condition</i> THEN <i>statement</i> <i>statement</i> ELSE <i>statement</i> <i>statement</i> END IF	IF num_dogs > 10 THEN DISPLAY "You have more than 10 dogs!" ELSE DISPLAY "You have ten or fewer dogs!" END IF
Use a switch/case statement	SELECT <i>variable or expression</i> CASE <i>value_1</i> : <i>statement</i> <i>statement</i> CASE <i>value_2</i> : <i>statement</i> <i>statement</i> CASE <i>value_2</i> :	SELECT num_dogs CASE 0: DISPLAY "No dogs!" CASE 1: DISPLAY "One dog.." CASE 2: DISPLAY "Two dogs.." CASE 3: DISPLAY "Three dogs.." DEFAULT: DISPLAY "Lots of dogs!" END SELECT

	<i>statement</i> <i>statement</i> DEFAULT: <i>statement</i> <i>statement</i> END SELECT	
<b>Loops</b>		
Loop while a condition is true - the loop body will execute 0 or more times.	WHILE <i>condition</i> <i>statement</i> <i>statement</i> END WHILE	SET num_dogs = 1 WHILE num_dogs < 10 DISPLAY num_dogs, " dogs!" SET num_dogs = num_dogs + 1 END WHILE
Loop while a condition is true - the loop body will execute 1 or more times.	DO <i>statement</i> <i>statement</i> WHILE <i>condition</i>	SET num_dogs = 1 DO DISPLAY num_dogs, " dogs!" SET num_dogs = num_dogs + 1 WHILE num_dogs < 10
Loop a specific number of times.	FOR <i>counter</i> = <i>start</i> TO <i>end</i> <i>statement</i> <i>statement</i> END FOR	FOR count = 1 TO 10 DISPLAY num_dogs, " dogs!" END FOR
<b>Functions</b>		
Create a function	FUNCTION <i>return_type</i> <i>name (parameters)</i> <i>statement</i> <i>statement</i> END FUNCTION	FUNCTION Integer add(Integer num1, Integer num2) DECLARE Integer sum SET sum = num1 + num2 RETURN sum END FUNCTION
Call a function	CALL <i>function_name</i>	CALL add(2, 3)
Return data from a function	RETURN <i>value</i>	RETURN 2 + 3