

# CS 161A: Programming and Problem Solving I

## Assignment xx 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:

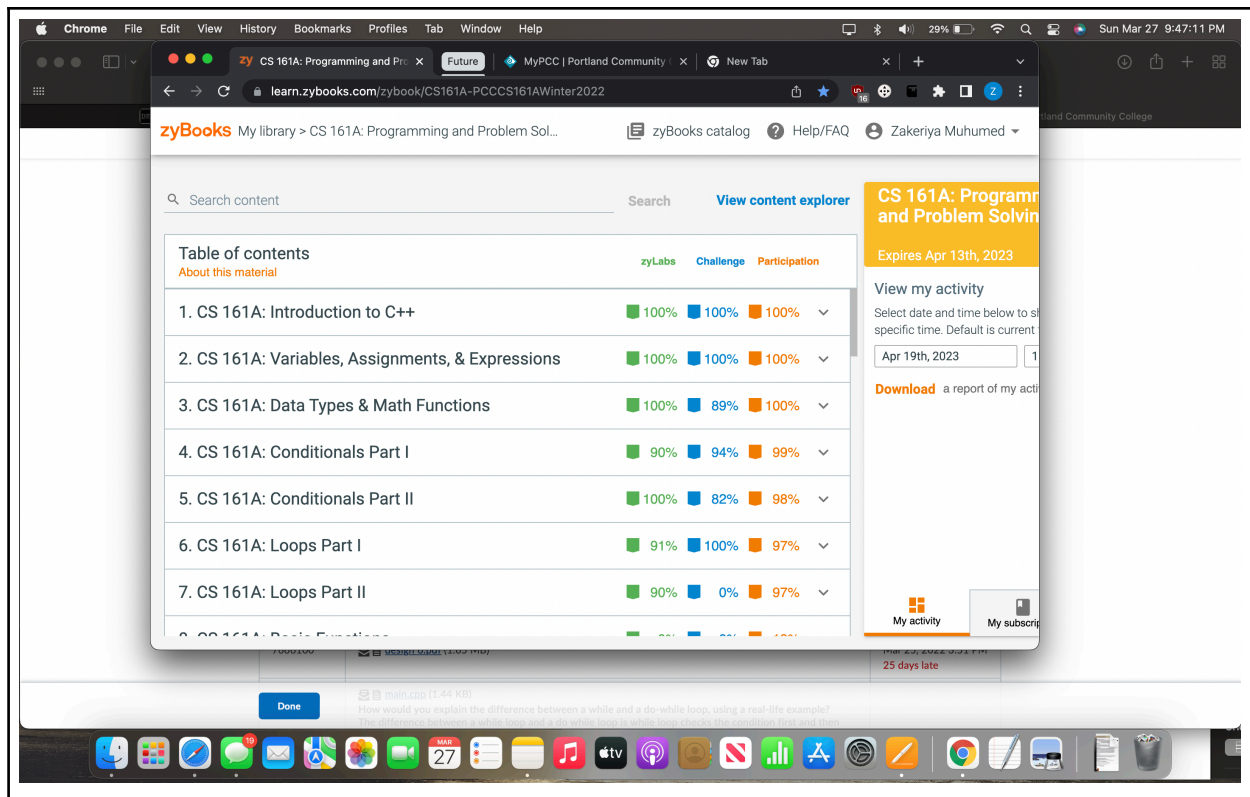
- ☐ Paste a screenshot of your zyBooks Challenge and Participation %
- ☐ Paste a screenshot of your assigned zyLabs completion
- ☐ Write a detailed description of your program, at least two complete sentences
- ☐ If applicable, design a sample run with test input and output
- ☐ Identify the program inputs 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](https://draw.io) - [Diagrams.net](https://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%.

<b>Challenge and Participation % screenshot:</b>
--



## 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:

In this assignment you will write an online ordering system for a coffee shop. Your program will continuously print a menu of options for the user to choose from and when they are done it will prompt for a tip amount and print the total pay amount.

## 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 my Coffee Shop!

Please pick an option below:
    1. Donuts/Muffins/Pastries
    2. Bagels/Toast
    3. Coffee/Tea
    4. Quit
>> c
Invalid Option! Please choose 1-4!
>> 9
Invalid Option! Please choose 1-4!
>> 1
Would you like
    D: Donuts $4.00
    M: Muffins $4.50
    P: Pastries $5.50
>> f
Invalid Option!
Would you like Donuts(D), Muffins (M), or Pastries (P): d
Donuts added.
Total: $4.00

Please pick an option below:
    1. Donuts/Muffins/Pastries
    2. Bagels/Toast
    3. Coffee/Tea
    4. Quit
>> 2
Would you like
    B: Bagels $3.75
    T: Toast $2.25
>> b
Bagels added.
Total: $7.75

Please pick an option below:
    1. Donuts/Muffins/Pastries
    2. Bagels/Toast
    3. Coffee/Tea
    4. Quit
>> 3
Would you like:
    C: Coffee $3.50
    T: Tea $2.50
>> t
Tea added.
Total: $10.25

Please pick an option below:
    1. Donuts/Muffins/Pastries
    2. Bagels/Toast
    3. Coffee/Tea
    4. Quit
>> 4

```

## 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.

<b>Algorithmic design:</b>
a. Identify and list all of the user input and their data types.
Cin type options Cin choose
b. Identify and list all of the user output and their data types.
Print Options, Choose, welcome message, thank you message, cost each time
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.
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.

prints the welcome message

prints the menu of the coffee house

reads the option selected by the user

executes the selected option

Store total

exits when user inputs 4:

which is associated with exit in the menu

Do statement excuting to read the menu option

While statement != to 4

Void for welcome message

Void for menu

Void for reading option

Void for options break

If statement for all the choose

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

