

# CS 161A: Programming and Problem Solving I

## Assignment 4 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

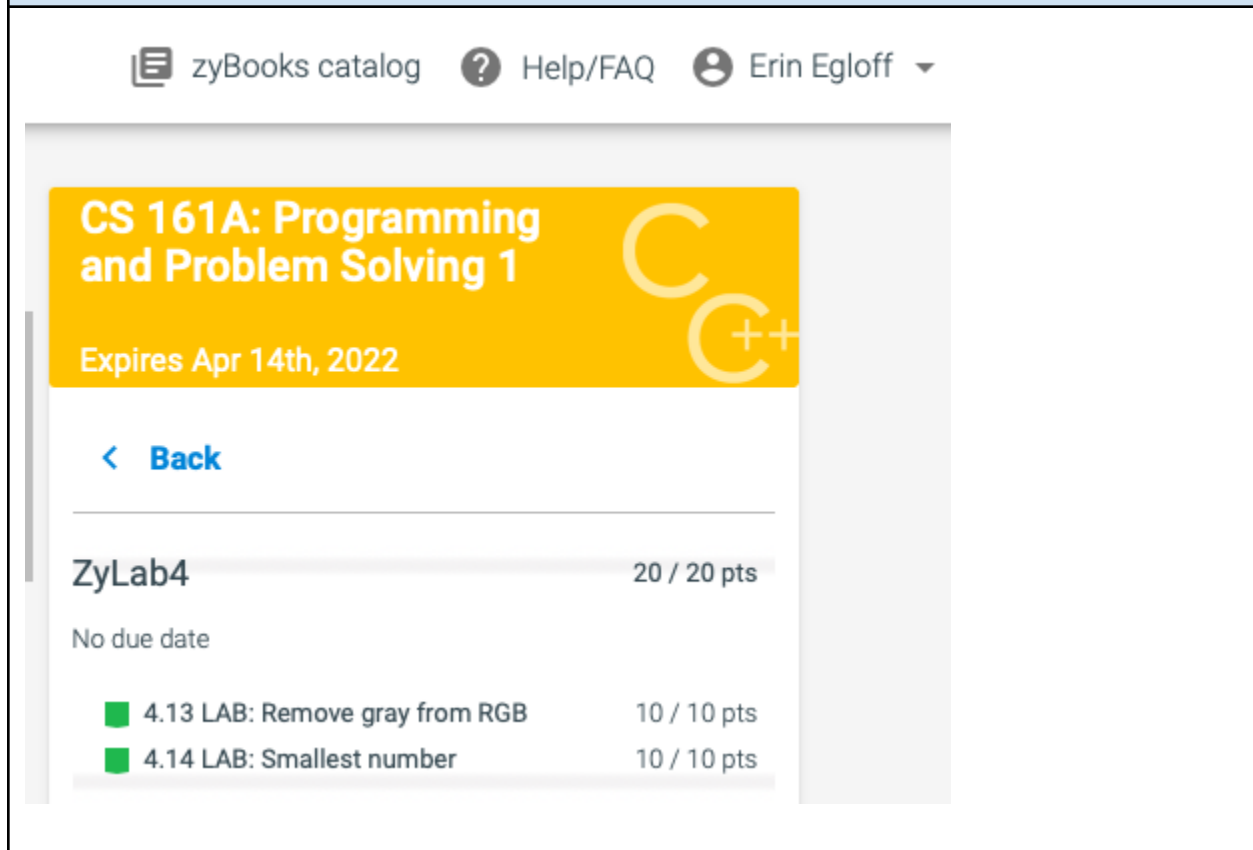
Add your zyBooks screenshots for the % and assigned zyLabs completions below. Required percentages: all **assigned** zyLabs, Challenge Activity with at least 70%, and Participation Activity with at least 80%.

### Challenge and Participation % screenshot:

The screenshot displays the zyBooks interface for the course 'CS 161A: Programming and Problem Solving 1'. The main content area shows a 'Table of contents' with four sections, each with a 'zyLabs', 'Challenge', and 'Participation' completion percentage. All sections show 100% completion. A sidebar on the right shows the course title, expiration date (Apr 14th, 2022), and a 'View my activity' section with a date and time selector.

	zyLabs	Challenge	Participation
1. CS 161A: Introduction to C++	100%	100%	100%
2. CS 161A: Variables, Assignments, & Expressions	100%	100%	100%
3. CS 161A: Data Types & Math Functions	100%	100%	100%
4. CS 161A: Conditionals Part I	100%	100%	100%

### Assigned zyLabs completion screenshot:



## 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 will calculate the total price based on their type of Hop Fastpass. The user will know the number of tickets purchased, how much it will cost, and the cost to qualify for free rides for the month (if applicable or not attained).

## 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 TriMet Hop FastPass!

FastPass Ticket	Ticket \$
-----------------	-----------

A. Adult Ticket (ages 18-64)	\$2.50
------------------------------	--------

H. Honored Citizen (65+, disabilities)	\$1.25
--	--------

Y. Youth Ticket (ages 7-17)	\$1.25
-----------------------------	--------

Note: Ride is 2 hours and 30 minutes with each ticket.

Ride free for the rest of the month after spending \$100 with an Adult pass, or \$28 with an Honored Citizen or Youth pass!

Please enter the type of pass you wish to purchase (A, H, Y):

(input letter)

Ex. h

Enter the number of tickets purchased this month:

(input number)

Ex. 20

You have purchased 20 tickets!

You have paid: \$ 25.00.

Spend \$ 3.00 more to earn free rides for the rest of the month.

Thank you for riding TriMet!

## 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.
Inputs: char typeFastPass, int numberTickets
b. Identify and list all of the user output and their data types.
Outputs: int totalCost, int needToSpend
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.
Cost for tickets  $\text{totalCost} = \text{typeFastPass} * \text{numberTickets}$  Amount needed to spend to earn free rides for the month  For adult pass: $\text{needToSpend} = \text{adultPassFree} - \text{totalCost}$  For honored citizen or youth pass: $\text{needToSpend} = \text{notAdultPassFree} - \text{totalCost}$
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.

```
DECLARE char typeFastPass, int numberTickets, double totalCost, double needToSpend,  
const double adultPassFree, const double nonAdultPassFree, const double adultTicket, const  
double nonAdultTicket
```

```
DISPLAY "Welcome to TriMet Hop Fastpass!"
```

```
DISPLAY FastPass Ticket:           Ticket $
```

```
A .Adult (ages 18-64)                $2.50
```

```
H. Honored Citizen (ages 65+, disabilities) $1.25
```

```
Y. Youth (ages 7-17)                 $1.25
```

```
DISPLAY "Note: Ride is 2 hours and 30 minutes ticket."
```

```
DISPLAY "Ride free for the rest of the month after spending $100 with an Adult pass, or $28  
with an Honored Citizen or Youth pass!"
```

```
DISPLAY "Enter FastPass (A, H, or Y): "
```

```
INPUT typeFastPass
```

```
    WHILE ((typeFastPass != 'A') && (typeFastPass != 'a') && (typeFastPass != 'H') &&  
(typeFastPass != 'h') && (typeFastPass != 'Y') && (typeFastPass != 'y'))
```

```
        DISPLAY "Bad Input! Please enter the letter A, H, or Y"
```

```
        INPUT typeFastPass
```

```
DISPLAY "Enter the number of tickets purchased this month: "
```

```
INPUT numberTickets
```

```
    WHILE (!cin)
```

```
        DISPLAY "Bad Input! Please enter the letter A, H, or Y"
```

```
        INPUT typeFastPass
```

```
DISPLAY "You have purchased " numberTickets " tickets!"
```

```
IF (typeFastPass == 'A') || (typeFastPass == 'a') {
```

```
    totalCost = adultTicket * numberTickets
```

```
    DISPLAY "You have paid: $" totalCost "."
```

```
}
```

```
ELSE IF (typeFastPass == 'H') || (typeFastPass == 'h') || (typeFastPass == 'Y') ||  
(typeFastPass == 'y') {
```

```
    totalCost = nonAdultTicket * numberTickets
```

```
    DISPLAY "You have paid: $" totalCost "."
```

```
}
```

```
IF (typeFastPass == 'A') || (typeFastPass == 'a') {
```

```
    needToSpend = adultPassFree - totalCost
```

```
    DISPLAY "Spend " needToSpend "more to earn free rides for the rest of the month."
```

```
}
```

```
ELSE IF (typeFastPass == 'H') || (typeFastPass == 'h') || (typeFastPass == 'Y') ||  
(typeFastPass == 'y') {
```

```
    needToSpend = nonAdultPassFree - totalCost
```

```
    DISPLAY "Spend " needToSpend "more to earn free rides for the rest of the month."
```

```
}
```

```
DISPLAY "Thank you for riding TriMet!"
```

## 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> : <i>statement</i> <i>statement</i> DEFAULT: <i>statement</i> <i>statement</i> END SELECT	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
<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