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
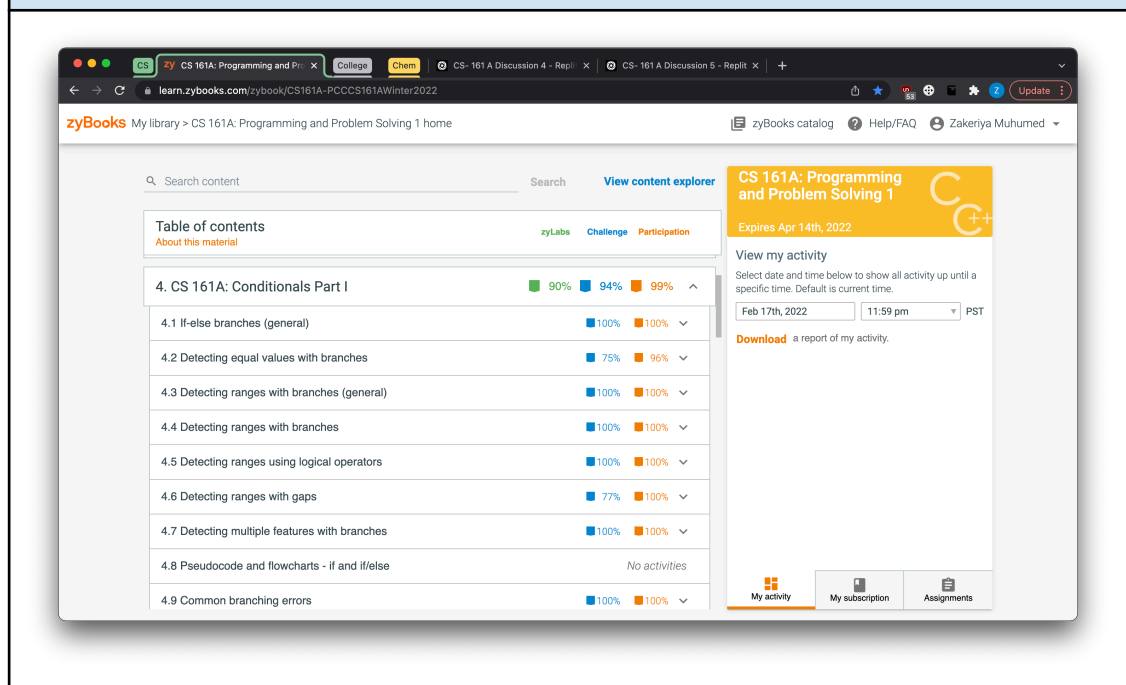- ❏ Tools for flowchart - Draw.io - Diagrams.net

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:
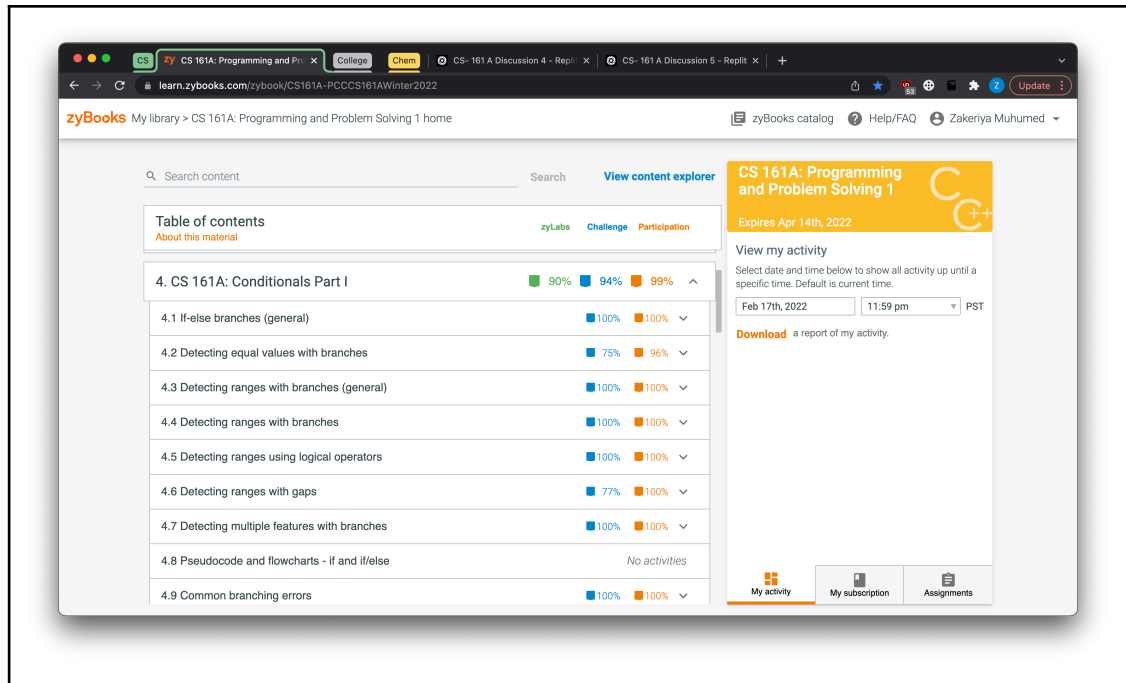
Assigned zyLabs completion screenshot:

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:
❏ prompt the user for the type of Hop Fastpass they purchase: Adult, Honored Citizen, or Youth (char) and the number of tickets they have purchased this month. Your program must be case insensitive (both 'A' and 'a' should work for Adult). ❏ Calculate the total price spent so far based on the number of tickets. Be careful, look at the table below for maximum costs, the total price should not exceed the maximum cost for riding free.

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 Choices                          Ticket $
--------------------------------------  --------
A. Adult (ages 18-64)                      $2.50
H. Honored Citizen (65+, disabilities)   $1.25
Y. Youth (ages 7-17)                       $1.25

Note: Ride for 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!

Enter Fastpass (A, H, Y): h

Enter the number tickets purchased this month: 1

You have purchased 1 ticket!
You have paid: $ 1.25
Spend $ 26.75 more to earn free rides for the rest of
the month!

Thank you for riding TriMet!
``` |

```
Welcome to TriMet Hop Fastpass!

Fastpass Choices                             Ticket $
------------------------------------  --------
A. Adult (ages 18-64)                         $2.50
H. Honored Citizen (65+, disabilities)   $1.25
Y. Youth (ages 7-17)                          $1.25

Note: Ride for 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!

Enter Fastpass (A, H, Y): A

Enter the number tickets purchased this month: 1

You have purchased 1 ticket!
You have paid: $ 2.50
Spend $ 97.50 more to earn free rides for the rest of
the month!

Thank you for riding TriMet!
```

Pass	Cost per ticket	Ride free after reaching this limit
Adult Ages 18 - 64.	$2.50	$100.00
Honored Citizen Seniors age 65+ and riders with disabilities.	$1.25	$28.00
Youth Ages 7 - 17.	$1.25	$28.00

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.
The fastpass type (A, H, Y):
Number tickets purchased this month

b. Identify and list all of the user output and their data types.
Display User welcome
Display Fastpass choices and ticket prices
Dispaly Note : ride for 2 hours and 30 mixtures with each tickets.
Display Ride for the rest of the month after spending $100 endline
Display with an Adult pass , or 28 with honors endline citizen or enplane
Display young pass !!
Display Enter fastpass type (A, H, Y):
DIsplay You have purchased numticket ticket!
Display you have paid spent month
Display spent '??' more to earn free ride for the rest of the month

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.
Fastpass {A = 100 H= 28 Y=28}
Prices {Adult = 2.50 Honors = 1.25 Youth = 1.25}
Paid = numtickets* price
Spent = fastpass - paid

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.

```
Display Welcome to TriMet Hop Fastpass!
```

DECLARE A, a, Y, y, H, h, numticket, numspent, fastpass, prices, numpaid,

Display User welcome

Display Fastpass choices and ticket prices Format

Dispaly Note : ride for 2 hours and 30 mixtures with each tickets.

Display Ride for the rest of the month after spending $100 endline

Display with an Adult pass , or 28 with honors endline citizen or enplane

Display young pass !!

Display Enter fastpass type (A, H, Y):

Input fastpass type (A, H, Y):

DIsplay You have purchased numticket ticket!

Input numtickets

IF(fastpass = A OR fastpass= h) {THEN

   SET A = 100

    Set prices = 2.50

    SET numpaid= prices*numticket

    SET numspend= A - paid

    Display You have purchased this month: num of ticket

    Display You have paid; numpaid;

   if numspent < 0

      Then display You have quality for free ride for the rest of the month.

  Else   Display Spend $ 'numspend' more to earn free rides for the rest of the mo[r]

   }

Else IF

 (fastpass = =H OR fastpass == h) {THEN

   SET A = 100

    Set prices = 1.25

    SET numpaid= prices*numticket

    SET numspend= H - paid

If numspent < 0

　　　Then display You have quality for free ride for the rest of the month.

　Else　　Display Spend $ 'numspend' more to earn free rides for the rest of the month!}

Ese IF

 fastpass = Y OR fast = y) {THEN

　SET Y = 100

　 Set prices = 1.25

　 SET numpaid= prices*numticket

　 SET numspend= Y - paid

　　 If numspent < 0

　　 Then display You have quality for free ride for the rest of the month.

　Else　　Display Spend $ 'numspend' more to earn free rides for the rest of the month!

　 Display You have purchased this month: num of ticket END IF

Else display invalid pass type

DIsplay

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`
Conditionals		

| Use a single alternative conditional | IF *condition* THEN
    statement
    statement
END IF | ```IF num_dogs > 10 THEN
    DISPLAY "That is a lot of
dogs!"
END IF``` |
|---|---|---|
| Use a dual alternative conditional | IF *condition* THEN
    statement
    statement
ELSE
    statement
    statement
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 *variable or expression*
  CASE *value_1:*
    statement
    statement
  CASE *value_2:*
    statement
    statement
  CASE *value_2:*
    statement
    statement
  DEFAULT:
    statement
    statement
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``` |

Loops

| Loop while a condition is true - the loop body will execute 0 or more times. | WHILE *condition*
    statement
    statement
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
    statement
    statement
WHILE *condition* | ```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 *counter* = *start* TO *end*
    statement
    statement
END FOR | ```FOR count = 1 TO 10
    DISPLAY num_dogs, " dogs!"
END FOR``` |

Functions

| Create a function | FUNCTION *return_type name (parameters)*
    statement
    statement
END FUNCTION | ```FUNCTION Integer add(Integer
num1, Integer num2)
    DECLARE Integer sum
    SET sum = num1 + num2
    RETURN sum
END FUNCTION``` |
|---|---|---|

Call a function	CALL *function_name*	`CALL add(2, 3)`
Return data from a function	RETURN *value*	`RETURN 2 + 3`