

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](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%.

Challenge and Participation % screenshot:

zyBooks My library > CS 161B: Programming II home

Search content

View content explorer

CS 161B: Programming II

Expires Jul 3rd, 2022

View my activity

Select date and time below to show all activity up until a specific time. Default is current time.

Apr 24th, 2022 11:59 pm PDT

Download a report of my activity.

My activity My subscription Assignments

Table of contents	zyLabs	Challenge	Participation
10.1 Pass by reference	100%	100%	
10.2 A Menu Driven Program - A Calculator			No activities
10.3 Scope of variable/function definitions		100%	
10.4 Default parameter values	100%	100%	
10.5 Function name overloading	100%	100%	
10.6 Parameter error checking		100%	
10.7 Preprocessor and include		100%	
10.8 Hand trace reference variables			No activities
10.9 Assignment Sample			No activities
10.10 LAB: Swapping variables	100%		
10.11 LAB: Flip a coin	100%		
10.12 LAB: Paint a room Optional	0%		

Assigned zyLabs completion screenshot:

zyBooks My library > CS 161B: Programming II home

Search content

View content explorer

CS 161B: Programming II

Expires Jul 3rd, 2022

View my activity

Select date and time below to show all activity up until a specific time. Default is current time.

Apr 24th, 2022 11:59 pm PDT

Download a report of my activity.

My activity My subscription Assignments

Table of contents	zyLabs	Challenge	Participation
10.1 Pass by reference	100%	100%	
10.2 A Menu Driven Program - A Calculator			No activities
10.3 Scope of variable/function definitions		100%	
10.4 Default parameter values	100%	100%	
10.5 Function name overloading	100%	100%	
10.6 Parameter error checking		100%	
10.7 Preprocessor and include		100%	
10.8 Hand trace reference variables			No activities
10.9 Assignment Sample			No activities
10.10 LAB: Swapping variables	100%		
10.11 LAB: Flip a coin	100%		
10.12 LAB: Paint a room Optional	0%		

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 program you will be writing a menu driven program to build a customized membership that works for the user. You will give the user a menu with some choices, and let them pick a choice. Based on the choice they pick, you will ask them some questions and give them results. This process will repeat until they choose to quit the program. The purpose of this assignment is to modularize your program.

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 fitClub program!!
The cost to become a member of the fitClub center is as follows:

The membership fee per month for Sports Club is $20.00

The membership fee per month for Ultra Sports Club is $30.00

If the membership is bought and paid for 12 or more months, the
discount is 10% off membership cost

Personal training sessions are $10.00 per session

If more than five personal training sessions are bought and paid
for, the discount on each session is 20%

Please pick one of the following options:

    (C/c) Calculate membership costs.
    (Q/q) quit this program.
```

>> c

Would you prefer (S)ports or (U)ltra Sports Club: f

Invalid membership type! Please try again!

Would you prefer (S)ports or (U)ltra Sports Club: s

How many months of membership would you like? 15

How many personal training sessions would you like? 0

Your membership cost after the 10% off is \$270.00

Your personal training cost for 0 sessions is \$0.00

Your total membership cost is \$270.00.

Please pick one of the following options:

(C/c) Calculate membership costs.

(Q/q) quit this program.

>> b

Invalid option!!

Please pick one of the following options:

(C/c) Calculate membership costs.

(Q/q) quit this program.

>> c

Would you prefer (S)ports or (U)ltra Sports Club: U

How many months of membership would you like? -15

Invalid number! Please try again!!!

How many months of membership would you like? 15

How many personal training sessions would you like? 10

Your membership cost after the 10% off is \$405.00

Your personal training cost for 10 sessions is \$80.00

Your total membership cost is \$485.00.

Please pick one of the following options:

(C/c) Calculate membership costs.

(Q/q) quit this program.

>> q

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.
<ul style="list-style-type: none">○ userChoice as char (to read the option from the user)○ clubType as char (to read the club type user wants)○ numMonth as integer (to read # month user wants membership)○ numTrain as integer (to read # of sessions)
b. Identify and list all of the user output and their data types.
<ul style="list-style-type: none">○ discount1 as double (to store the discount user receives from month)○ discount2 as double (to store the discount user receives from alot of session)○ totalCost as double (to store the total cost)○ membershipCost as double (to store the cost for clubType)○ TrainCost as double(to store the total cost of training)
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">○ SET discount2 = numTrain*0.20○ SET trainCost = numTrain - discount2○ SET Discount1 = membershipCost * 0.10○ SET membershipCost = membershipCost - discount;○ SET membershipCost = numMonth * 20○ SET membershipCost = numMonth * 30
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.
1. FUNCTION welcome() a. Welcome to my fitClub program!!

The cost to become a member of the fitClub center is as follows:

The membership fee per month for Sports Club is \$20.00

The membership fee per month for Ultra Sports Club is \$30.00

If the membership is bought and paid for 12 or more months, the discount is 10% off membership cost

Personal training sessions are \$10.00 per session

If more than five personal training sessions are bought and paid for, the discount on each session is 20%

END FUNCTION welcome()

2. FUNCTION readInt(string prompt) (Data Validation function for an integer)

- a. DECLARE tempInt as int
- b. DISPLAY prompt
- c. INPUT into tempInt
- d. WHILE tempVar not an int
 - i. DISPLAY error message
 - ii. RESET input stream
 - iii. CLEAR buffer
 - iv. INPUT into tempInt
- e. END WHILE
- f. return tempInt

END FUNCTION readInt()

3. FUNCTION displayMenu()

- a. Please pick one of the following options:
(C/c) Calculate membership costs.
(Q/q) quit this program.

END FUNCTION displayMenu()

4. FUNCTION readChar(string prompt) (Data Validation function for an integer)

- a. DECLARE tempInt as char
- b. DISPLAY prompt
- c. INPUT into CAP tempChar
- d. WHILE tempVar not input
 - i. DISPLAY error message
 - ii. RESET input stream
 - iii. CLEAR buffer
 - iv. INPUT into CAP tempChar
- e. END WHILE

f. return tempChar

END FUNCTION readChar()

5. **FUNCTION getInput**(double &clubType, double &numMonth, double &numTrain)

a. **DO**

i. **CALL** readChar function to read clubType

b. **WHILE** clubType is not S or U

c. **CALL** readInt function to read numMonth

d. **CALL** readInt function to read numTrain

END FUNCTION getInput()

6. **FUNCTION calcCost**(clubType, numMonth, numTrain, totalCost)

a. **DECLARE** discount 1 as double

b. **DECLARE** discount 2 as double

c. **DECLARE** membershipCost as double

d. **DECLARE** trainCost = numTrain * 10

e. **SELECT** uppercase clubType

i. **CASE** 'S':

1. **SET** membershipCost = numMonth * 20

ii. **CASE** 'U':

1. **SET** membershipCost = numMonth * 30

f. **IF** numTrain > 5

i. **SET** discount2 = numTrain * 0.20

ii. **SET** trainCost = numTrain - discount2

g. **END IF**

h. **IF** numMonth >= 12

i. **SET** Discount1 = membershipCost * 0.10

ii. **SET** membershipCost = membershipCost - discount;

iii. **DISPLAY** "your membership cost with 10% off is" membershipCost

i. **END IF**

j. **SET** totalCost = membershipCost + trainCost

k. **DISPLAY** number of training: numTrain

l. **DISPLAY** total training cost : trainCost

m. **DISPLAY** total membership cost: totalCost

END FUNCTION calcCost()

7. **FUNCTION main()**

a. **DECLARE** userChoice as char

b. **DECLARE** numTrain as int

c. **DECLARE** numMonth as int

d. **DECLARE** totalCost as as double

e. **CALL** welcome()

```

f. DO LOOP
    i. CALL displayMenu()
    ii. DO LOOP
        1. CALL readChar and get userChoice
    iii. WHILE userChoice is not C OR Q
    iv. SELECT uppercase userChoice
        1. CASE 'C':
            a. CALL getInput(numMonths, clubType, numTrain)
            b. CALL calCost(numMonths, clubType, numTrain,
                totalCost)
        2. CASE 'Q':
            a. DISPLAY Thank you, see you next time message
    v. END SELECT
g. END DO WHILE userChoice != 'Q'
8. END FUNCTION main()

```

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 <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> CASE <i>value_2</i> : <i>statement</i> CASE <i>value_2</i> : <i>statement</i> CASE <i>value_2</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
Loops		
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
Functions		
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