

CS 161A: Programming and Problem Solving I

Assignment A04 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:



☐ 4. CS 161A: Conditionals Part I ■ 100% ■ 100% ■ 100% ^

Assigned zyLabs completion screenshot:

<input type="checkbox"/>	4.14 LAB: Remove gray from RGB	100%	▼
<input type="checkbox"/>	4.15 LAB: Smallest number	100%	▼

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 fare due to the Washington state ferry service. Simply input if you intend to use a vehicle on the ferry or not alongside the total number of people in your party.

This will output the total fare alongside any discounts you may receive, and check if the number of people in your party is allowed.

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 the Washington State Ferries Fare Calculator!

Fare Description	Ticket \$
Vehicle Under 14' (less than 168") & Driver	\$57.90
Adult (age 19 - 64)	\$14.95
Senior (age 65 & over) / Disability	\$7.40
Youth (age 6 - 18)	\$5.55
Bicycle Surcharge (included with Vehicle)	\$4.00

Are you riding a vehicle on the Ferry (Y/N): **Y**

How many adults? **5**

How many seniors? **0**

How many youths? 7

Your total charge is \$171.50

You have spent \$100 or more. You have received a free adult ticket for your next trip

Thank you for using Washington State Ferries Fare Calculator!

Welcome to the Washington State Ferries Fare Calculator!

Fare Description	Ticket \$
Vehicle Under 14' (less than 168") & Driver	\$57.90
Adult (age 19 - 64)	\$14.95
Senior (age 65 & over) / Disability	\$7.40
Youth (age 6 - 18)	\$5.55
Bicycle Surcharge (included with Vehicle)	\$4.00

Are you riding a vehicle on the Ferry (Y/N): y

How many adults? 5

How many seniors? -2

Error!! Invalid answer!! Please try again later!!!

Thank you for using Washington State Ferries Fare Calculator!

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.

Int numAdults

Int numSeniors

Int numYouths

Int numBikes

String usingFerry

b. Identify and list all of the user output and their data types.

Double fareTotal

Double 100 - fareTotal

String errorMessage

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.

fareTotal = fareTotal + (numAdults * FARE_ADULT);

fareTotal = fareTotal + (numSeniors * FARE_SENIOR);

fareTotal = fareTotal + (numYouths * FARE_YOUTH);

fareTotal = fareTotal + (numBikes * FARE_BIKE);

[Check remainder of 100 from fare] 100 - fareTotal;

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.

START

Welcome to the Washington State Ferries Fare Calculator!

Fare Description	Ticket \$
-----	-----
Vehicle Under 14' (less than 168") & Driver	\$57.90
Adult (age 19 - 64)	\$14.95
Senior (age 65 & over) / Disability	\$7.40
Youth (age 6 - 18)	\$5.55
Bicycle Surcharge (included with Vehicle)	\$4.00

Are you riding a vehicle on the Ferry (Y/N): y

How many adults? 2

How many seniors? 1

How many youths? 0

Your total charge is \$95.20

If you spend \$4.80 more, you are eligible for a free adult ticket for the next trip.

START

DECLARE GLOBAL Double VEHICLE_168 = 57.90

DECLARE GLOBAL Double FARE_ADULT = 14.95

DECLARE GLOBAL Double FARE_SENIOR = 7.40

DECLARE GLOBAL Double FARE_YOUTH = 5.55

DECLARE GLOBAL Double FARE_BIKE = 4.00

DECLARE Int numAdults = 0

DECLARE Int numSeniors = 0

DECLARE Int numYouths = 0

DECLARE Int numBikes

DECLARE Double fareTotal = 0.00

DECLARE Str usingFerry = "init"

DECLARE Int errorState = 0

SET maximum decimal output = 2

DISPLAY "Welcome to the Washington State Ferries Fare Calculator!"

DISPLAY "Fare Description	Ticket \$"
---------------------------	------------

DISPLAY "-----"	-----"
-----------------	--------

DISPLAY "Vehicle Under 14' (less than 168") & Driver	\$57.90"
--	----------

DISPLAY "Adult (age 19 - 64)	\$14.95"
------------------------------	----------

DISPLAY "Senior (age 65 & over) / Disability	\$7.40"
--	---------

DISPLAY "Youth (age 6 - 18)	\$5.55"
-----------------------------	---------

DISPLAY "Bicycle Surcharge (included with Vehicle)	\$4.00"
--	---------

DISPLAY "Are you riding a vehicle on the Ferry (Y/N): "

INPUT usingFerry

```

IF usingFerry != "n" OR needFerry != "N" OR needFerry != "y" OR needFerry != "Y"
-- SET errorState = 1
ELSE IF usingFerry = "y" OR usingFerry = "Y"
-- SET fareTotal = VEHICLE_168

IF errorState != 1
-- DISPLAY "How many adults? "
-- INPUT numAdults
-- IF numAdults < 0
---- SET errorState = 1

IF errorState != 1
-- DISPLAY "How many seniors? "
-- INPUT numSeniors
-- IF numSeniors < 0
---- SET errorState = 1

IF errorState != 1
-- DISPLAY "How many youths? "
-- INPUT numYouths
-- IF numYouths < 0
---- SET errorState = 1

IF errorState != 1
-- IF usingFerry == "n" OR usingFerry == "N"
---- DISPLAY "How many Bikes? "
---- INPUT numBikes
---- IF numBikes < 0
----- SET errorState = 1

IF errorState == 1
-- DISPLAY "Error!! Invalid answer!! Please try again later!!!"
ELSE IF (numAdults + numSeniors + numBikes) > 21
-- DISPLAY "Uh oh!! Too many people in your group. Split into 2 groups and try again!"
ELSE
-- SET fareTotal = fareTotal + (numAdults * FARE_ADULT)
-- SET fareTotal = fareTotal + (numSeniors * FARE_SENIOR)
-- SET fareTotal = fareTotal + (numYouths * FARE_YOUTH)
-- SET fareTotal = fareTotal + (numBikes * FARE_BIKE)
--

```

```
- - DISPLAY "Your total charge is ${fareTotal}"
- - DISPLAY "If you spend ${100 - fareTotal} more, you are eligible for a free adult ticket for
the next trip."
```

```
DISPLAY "Thank you for using Washington State Ferries Fare Calculator!"
```

```
END
```

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

	CASE <i>value_2</i> : <i>statement</i> <i>statement</i> DEFAULT: <i>statement</i> <i>statement</i> 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