

## 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 and submit with your Assignment to D2L (File -> Download -> PDF). The sections will expand as you type.

### 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%.

#### zyLabs, Challenge, and Participation % Screenshot:

[About this material](#)

1. Introduction to C

100% 100% 100% ▾

#### Assigned zyLabs completion Screenshot:

1.16 zyLab training: Basics

100% ▾

1.17 zyLab training: Interleaved input / output

100% ▾

1.18 LAB: Formatted output: Hello World!

100% ▾

1.19 LAB: Formatted output: No parking sign

100% ▾

1.20 LAB: Input and formatted output: Right-facing arrow

100% ▾

### Assignment

#### Program description:

This program will take the number of large and small pizza's you are ordering, calculate how much the total order will be, and provide an itemized cost breakdown.

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 all of the user input. What are the data types of the inputs? Define the input variables.

We will require 2 inputs;

1. A whole number of large pizza(s) intended to be purchased
2. A whole number of small pizza(s) intended to be purchased

b. Describe the program output. What is displayed to the user? What are the data types of the output? Define the output variables.

The program will output an arithmetic sum calculated from the supplied input values

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.

This calculation will require both multiplication and addition.

For safe coding practice, these arithmetic operations will need to be executed in series

d. Design the logic of your program using pseudocode or flowcharts. See pseudocode syntax at the bottom of this document. Here is where you would use conditionals, loops, functions or array constructs (if applicable) and list the steps in transforming inputs into outputs. Walk through your logic steps with the test data from the assignment document.

START

DECLARE Int costSmall = 5

DECLARE Int costLarge = 9

DECLARE Int numSmall = 0

DECLARE Int numLarge = 0

DECLARE Int sumTotal = 0

DISPLAY "Hello! Welcome to the pizza cost calculator."

DISPLAY "Please input the number of small pizza's being ordered: "

INPUT Int numSmall

DISPLAY "Please input the number of large pizza's being ordered: "

INPUT Int numLarge

SET Int sumTotal = (numSmall \* costLarge) + (numSmall \* costLarge)

DISPLAY "You are ordering ", numSmall, " small pizzas"

DISPLAY "You are ordering ", numLarge, " large pizzas"

DISPLAY "Order total: \$", sumTotal

DISPLAY numSmall, " pizza(s) at \$5/unit is: ", (numSmall \* costSmall)

DISPLAY numLarge, " pizza(s) at \$9/unit is: ", (numLarge \* costLarge)

DISPLAY "Thank you for using this calculator! Please report any bugs found."

END

- e. Include 2 Sample Program Runs for your program using your own set of data. This data set must be different from my Sample Runs in the Assignment document. This process is similar to Unit Testing and will help you test your program better.

Sample Program Run 1:

Hello! Welcome to the pizza cost calculator.

Please input the number of small pizza's being ordered: **5**

Please input the number of large pizza's being ordered: **3**

Order total: \$52

-----

\$5 dollar small pizza x **5** is: \$25

\$9 dollar large pizza x **3** is: \$27

Thank you for using this program! Please report any bugs found.

Sample Program Run 2:

Hello! Welcome to the pizza cost calculator.

Please input the number of small pizza's being ordered: **0**

Please input the number of large pizza's being ordered: **9**

Order total: \$81

-----

\$5 dollar small pizza x **0** is: \$0

\$9 dollar large pizza x **9** is: \$81

Thank you for using this program! Please report any bugs found.

## 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

Functions		
Create a function	FUNCTION <i>return_type</i> <i>name (parameters)</i> <i>statement</i> <i>statement</i> END FUNCTION	<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