CS 161A: Programming and Problem Solving I

Final 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

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:

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 Miles Tracker program.

How many miles do you want to ride this week? 50

How many miles did you ride on Sunday? -9
Miles must be 0 or greater!
How many miles did you ride on Sunday? -3
Miles must be 0 or greater!
How many miles did you ride on Sunday? 0
How many miles did you ride on Monday? 10
How many miles did you ride on Tuesday? 10
How many miles did you ride on Wednesday? 10
How many miles did you ride on Thursday? 10
How many miles did you ride on Friday? 10
How many miles did you ride on Saturday? 0

You rode 50 miles this week.
Good job! You met your goal!

Keep riding!
```

```
How many miles do you want to ride this week? 100

How many miles did you ride on Sunday? 10
How many miles did you ride on Monday? 10
How many miles did you ride on Tuesday? 10
How many miles did you ride on Wednesday? 10
How many miles did you ride on Thursday? 10
How many miles did you ride on Friday? 10
How many miles did you ride on Saturday? 10

You rode 70 miles this week.
Uh oh! You missed your goal by 30 miles!

Keep riding!
```

```
How many miles do you want to ride this week? 50

How many miles did you ride on Sunday? 10
How many miles did you ride on Monday? 10
How many miles did you ride on Tuesday? 10
How many miles did you ride on Wednesday? 10
How many miles did you ride on Thursday? 10
How many miles did you ride on Friday? 10
How many miles did you ride on Saturday? 10

You rode 70 miles this week.
Great job! You exceeded your goal by 20 miles!

Keep riding!
```

```
How many miles do you want to ride this week? 0

No miles were tracked this week.

Keep riding!
```

```
How many miles do you want to ride this week? -10

No miles were tracked this week.

Keep riding!
```

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.
Cin desireMiles Cin milesSun Cin milesMon Cin milesTue Cin milesWed Cin milesThur Cin. milesFri Cin milesSat
b. Identify and list all of the user output and their data types.

Print

Cout “How many miles do you want to ride this week? “ Endline end

Cout “How many miles did you ride on Sunday “ endline

Cout “How many miles did you ride on Monday “ endline

Cout “How many miles did you ride on Tuesday “ endline

Cout “How many miles did you ride on Wednesday “ endline

Cout “How many miles did you ride on Thursday “ endline

Cout “How many miles did you ride on Friday “ endline

Cout “How many miles did you ride on Saturday “ endline endline

Cout “You rode sum miles this week.”

Cout reaction “You exceeded or missed your goal by goal - all mile input

Print react and sub and sum

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.

sum

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 Int desireMiles

Declare Int milesSun

Declare Int milesMon

Declare Int milesTue

Declare Int milesWed

Declare Int milesThur

Declare Int milesFri

Declare Int milesSat Declare desireMiles - sum;

Sum= milesSun +milesMon+milesTue+milesWed + milesThur + milesFri + milesSat;

Do statement Print Cout “How many miles do you want to ride this week? “ Endline endline

Input desireMiles with a while statement of desireMiles < 0;

Print Cout “How many miles did you ride on Sunday “ endline

Input milesSun While loop to reenter for number under 0 end when value > 0

Print Cout “How many miles did you ride on Monday “ endline

Input milesMon While loop to reenter for number under 0 end when value > 0

Print Cout “How many miles did you ride on Tuesday “ endline

Input milesTue While loop to reenter for number under 0 end when value > 0

Print Cout “How many miles did you ride on Wednesday “ endline

Input milesWed While loop to reenter for number under 0 end when value > 0

Print Cout “How many miles did you ride on Thursday “ endline

Input milesThur While loop to reenter for number under 0 end when value > 0

Print Cout “How many miles did you ride o n Friday “ endline

Input milesFri While loop to reenter for number under 0 end when value > 0

Print Cout “How many miles did you ride on Saturday “ endline

Input milesSat While loop to reenter for number under 0 end when value > 0

Print cout “ You rode “ sub ‘“ miles this week.”

Print Reaction case! If else stamens for reaction Print “You missed or exceed your

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`