

# CS 160: Exploring Computer Science

## Algorithm 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 it to D2L.

This document contains an interactive checklist. To mark an item as complete, click on the box.

Planning your program before you start coding is part of the development process. In this document, you will:

- ☒ ~~Step 1: Write a detailed description of your program, at least two complete sentences~~
- ☒ ~~Step 2: If applicable, design a sample run with test input and output~~
- ☐ Step 3: Algorithm design
  - ☒ ~~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. Look at the Pseudocode syntax at the bottom of this document. Tools for flowchart [Draw.io](#) [Diagrams.net](#)~~

### 1. 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:

My program will calculate the time it will take to travel a certain amount of distance at a certain speed. The user will input their distance and their average speed, then the program will calculate the estimated time to travel that distance.

### 2. 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:

How far is the destination? (In miles): 10

What is your traveling speed? (In MPH): 60

You will arrive at your destination in 10 minutes

### 3. 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.

Use the pseudocode syntax shown in the document, supplemented with English phrases if necessary. **Do not include any implementation details (e.g. source code file names, or language syntax).** Do not include any Python specific syntax or data types.

#### Algorithmic design:

- a. Identify and list all of the user input and their data types. Include a variable name, data type, and description. Simple data types include string, integer, floating point, (single) character, and boolean. Complex Data structures like lists should be referenced by name, e.g. "array of integer" or "array of string".

Integer "distance"

Integer "speed"

- b. Identify and list all of the user output and their data types. Include a variable name, data type, and description. Data types include string, integer, floating point, (single) character, and boolean. Complex Data structures like lists should be referenced by name, e.g. "array of integer" or "array of string".

Float "time" (In minutes)

- 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. Formulae should reference the variable names from step a and step b as applicable.

$\text{time} = (\text{distance} / \text{speed}) * 60$

- 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. **Use the syntax shown at the bottom of this**

**document and plain English phrases. Do not include any implementation details (e.g. file names) or Python or any language specific syntax.**

INPUT distance

INPUT speed

DECLARE float time

SET time = (distance / speed) \* 60

DISPLAY "You will arrive at your destination in " + time + " minutes"

## 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>	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	
<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
<b>Functions</b>		
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