

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:

4. Loops

100% 100% 100% ▼

Assigned zyLabs completion Screenshot:

4.12 C example: Salary calculation with loops

No activities

4.13 LAB: Varied amount of input data

100% ▼

4.14 LAB: Countdown until matching digits Optional

0% ▼

4.15 LAB: Output range with increment of 5

100% ▼

Assignment

Program description:

This program will estimate the total number of jellybeans in a jar when given the jellybean size and jar size. Program will also show the jar size that held the largest amount of jellybeans, the average number of Jellybeans, and which jellybean input had the greatest volume!

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:

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

We have two input floats entered simultaneously in one input statement.

"[float][float]" userInput

userInput is split using a space and entered left-right

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 a series of print statements, some with data, some without. All data outputed is an integer or float rounded to .01. Outputs are as follows:

Welcome to my estimator for Jellybeans-in-the-jar!

Please enter the Length and Height of the Jellybean in CM.

Data must be separated by a space (ie: 2.1 3.4):

Given Jellybean height: [float]

Given Jellybean width: [float]

Please enter the jar size in ml:

Given jar volume: [integer]

Estimated number of beans in the jar: [integer]

Would you like to run this program again? (Yes/No):

Total entries given: [float]

The average number of Jellybeans was: [float]

The biggest Jellybean was: [float]cm^3

The jar size for the largest Jellybean est. was: [integer]

Thanks for using this calculator!

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.

beanVol = (5 * PI * beanHeight * pow(beanWidth, 2)) / 24;

beanEst = (jarVol * LOAD_FACTOR) / beanVol;

beanTotal = beanTotal + beanEst;

runCount = runCount + 1

beanTotal / runCount

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 Const Double PI = 3.14159265358979323846

DECLARE Const Double LOAD_FACTOR = 0.698

DECLARE Double beanHeight = 0.0

DECLARE Double beanWidth = 0.0

DECLARE Double beanVol = 0.0

DECLARE Double beanVolMax = 0.0

DECLARE Int beanEst = 0

DECLARE Double beanEstMax = 0.0

DECLARE Double beanTotal = 0

DECLARE Int jarVol = 0

DECLARE Int jarVolMax = 0

DECLARE Char userInputStr[256] = ""

DECLARE Double runCount = 0.0

SET userInputStr[0] = "y"

DISPLAY "Welcome to my estimator for Jellybeans-in-the-jar!"

WHILE userInputStr[0] = "y" OR userInputStr[0] = "Y"

- - SET beanHeight = 0.0

- - SET beanWidth = 0.0

- - SET beanEst = 0.0

- -

- - DISPLAY "Please enter the Length and Height of the Jellybean in CM."

- - DISPLAY "Data must be seperated by a space (ie: 2.1 3.4):

- -

- - INPUT beanHeight, beanWidth

- -

- - DISPLAY "Given Jellybean height: " + beanHeight

- - DISPLAY "Given Jellybean width: " + beanWidth

- -

- - DISPALY "Please enter the jar size in ml: "

- - INPUT jarVol

- - DISPLAY "Given jar volume: " + jarVol

- -

- - SET beanVol = (5 * PI * beanHeight * (beanWidth^2))

- - SET beanEst = ((jarVol * LOAD_FACTOR) / beanVol)

- - SET beanTotal = beanTotal + beanEst

- - DISPLAY "Estimated number of beans in the jar: " + beanEst

- -

- - IF beanVol > beanVolMax

```

- - - SET beanVolMax = beanVol
- -
- - IF beanEst > beanEstMax
- - - SET beanEstMax = beanEst
- - - SET jarVolMax = jarVol
- -
- - DISPLAY "Would you like to run this program again? (Yes/No): "
- - INPUT userInputStr
- - SET runCount = runCount + 1

DISPLAY "Total entries given: " + runCount
DISPLAY "The average number of Jellybeans was: " + (beanTotal / runCount)
DISPLAY "The jar size for the largest Jellybean est. was: " + beanVolMax + "cm^3"
DISPLAY "The jar size for the largest Jellybean est. was: " + jarVolMax + "ml"

DISPLAY "Thanks for using this calculator!"

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:

Welcome to my estimator for Jellybeans-in-the-jar!

Please enter the Length and Height of the Jellybean in CM.

Data must be separated by a space (ie: 2.1 3.4): **1.8 1.2**

Given Jellybean height: 1.80

Given Jellybean Width: 1.20

Please enter the jar size in ml: **750**

Given jar volume: 750

Estimated number of beans in the jar: 308

Would you like to run this program again? (Yes/No): **n**

Total entries given: 1

The average number of Jellybeans was: 308.00

The biggest Jellybean was: 1.70 cm³

The jar size for the largest Jellybean est. was: 750ml

Thanks for using this calculator!

Sample Program Run 2:

Welcome to my estimator for Jellybeans-in-the-jar!

Please enter the Length and Height of the Jellybean in CM.

Data must be separated by a space (ie: 2.1 3.4): **1.3 2.0**

Given Jellybean height: 1.30

Given Jellybean Width: 2.00

Please enter the jar size in ml: **491**

Given jar volume: 491

Estimated number of beans in the jar: 100

Would you like to run this program again? (Yes/No): y

Please enter the Length and Height of the Jellybean in CM.

Data must be separated by a space (ie: 2.1 3.4): **2.1 .5**

Given Jellybean height: 2.10

Given Jellybean Width: 0.50

Please enter the jar size in ml: **300**

Given jar volume: 300

Estimated number of beans in the jar: 609

Would you like to run this program again? (Yes/No): y

Please enter the Length and Height of the Jellybean in CM.

Data must be separated by a space (ie: 2.1 3.4): **6.1 9.2**

Given Jellybean height: 6.10

Given Jellybean Width: 9.20

Please enter the jar size in ml: **1500**

Given jar volume: 1500

Estimated number of beans in the jar: 3

Would you like to run this program again? (Yes/No): **No:)**

Total entries given: 3

The average number of Jellybeans was: 237.33

The biggest Jellybean was: 337.92 cm³

The jar size for the largest Jellybean est. was: 300ml

Thanks for using this calculator!

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> 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 |
| 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 |