# **Assignment 1 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:** |
|  |

|  |
| --- |
| **Assigned zyLabs completion Screenshot:** |
|  |

## Assignment

|  |
| --- |
| **Program description:** |
| This program will accept input from a user for a jelly bean’s height and length, as well as the volume of a jar. The program will first calculate the volume of a single jelly bean, and then calculate the number of jelly beans that will fit into a given jar volume and output this value back to the user. |

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:** |
| 1. Identify all of the user input. What are the data types of the inputs? Define the input variables. |
| **Variable Name / Description / Input Data Type**   1. beanLength / Length of jelly bean / float 2. beanHeight / Height of jelly bean / float 3. volumeOfJar / Volume of jar / int |
| 1. Describe the program output. What is displayed to the user? What are the data types of the output? Define the output variables. |
| The output will first greet the user, then ask for input, then output the number of jelly beans that will fit in the jar volume given.   1. A welcome message 2. Prompt asking the user for jelly bean length and height, and volume of jar. 3. Output telling the user an estimate of the number of jelly beans in the jar.   Output Variables:   1. Welcome message: string 2. Prompt for input: string 3. Output estimate: string, integer |
| 1. 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. |
| Formula for the volume of a single jelly bean:  **( 5 \* PI \* beanLength \* beanHeight ^ 2 ) / 24**  Formula to calculate number of beans that fit in the given jar volume:  **volumeOfJar \* LOADFACTOR / volumeOfSingleBean** |
| 1. Design the logic of your program using pseudocode or flowcharts. 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. |
| 1. Accept input from user: jelly bean length and height, jar volume 2. Calculate single jelly bean volume:    1. ( 5 \* PI \* beanLength \* beanHeight ^ 2 ) / 24   3. Calculate number of beans that fit in given jar volume:   * 1. volumeOfJar \* LOADFACTOR / volumeOfSingleBean |
| 1. 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:  (One ml equals one cubic centimeter)  ( 5 \* PI \* beanLength \* beanHeight ^ 2 ) / 24  beanLength: 1.75  beanHeight: .35  volumeOfJar: 350  beanHeight^2 = 0.1225  ( 5 \* 3.14159265358979323846 \* 1.75 \* .1225 ) / 24 = .140308 (volume of one)  (350 \* .745) / .140308 =  1858 beans  Sample Program Run 2:  ( 5 \* PI \* beanLength \* beanHeight ^ 2 ) / 24  beanLength: 1.25  beanHeight: .6  volumeOfJar: 275  beanHeight^2 = .6  ( 5 \* 3.14159265358979323846 \* 1.25 \* .36 ) / 24 = .2945 (volume of one)  (275 \* .745) / .2945 =  695 beans |