Assignment 1 Algorithmic Design Document

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:
Collect from the user an **ordered list** of three variables: `[jellyBeanLength (cm) | jellyBeanHeight (cm) | jarVolume (mL)]` Then write a calculation to predict the max number of jelly beans that can fit in the jar described by user input. Lastly, output the data for display according to specification: • Follow all req's for a01.cpp specifications • Loop the user input and calculations to display the estimate of max jelly beans in each jar until the user inputs "0 0 0" to exit • Then output the avg beans for all entries and the bean/jar dimensions for the

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.

```
double jellyBeanLength;
double jellyBeanHeight;
unsigned int jarVolume;
```

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

```
Announce user input request, accept user data. Then output accurate
prediction of integer value maxBeans in the jar, dropping the
fractional portion of the estimate.
```

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.

```
maxBeans = jarVolume * LOADFACTOR / beanVolume;
beanVolume = (5 * piVal * beanLength * beanHeight) / 24
```

d. 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.

LOAD libraries
DECLARE variables
COLLECT user input [cin >> length >> height >> jarVolume
CALCULATE output variables:

- beanVolume= 5 * PI * beanLength * (beanHeight ^2) / 24
- maxBeans = jarVolume * LOADFACTOR / beanVolume
- totalBeans += maxBeans

TEST IF maxBeans > mostBeans
THEN UPDATE mostBeans, bestLength, bestHeight, bestJar
TEST IF user entry != “0 0 0”
THEN UPDATE counter for number of entries and
DISPLAY estimate of jelly beans in the current jar
LOOP to COLLECT step until COLLECT = “0 0 0”
DISPLAY number of entries, exclude “0 0 0”
DISPLAY avg beans for all entries (totalBeans / entries)
DISPLAY best dimensions for bean and jar for biggest estimate
DISPLAY exit salutation