Lab Partner 1 Anwar Adam

Lab Partner 2 Alan Palayil

**CS 115 Fall 2019 Lab #1**

Due: **Tuesday, September 3rd, 5:00 PM**

Points: **20**

**Instructions:**

1. Use this document template to report your answers. Enter all lab partner names at the top of first page.
2. You don’t need to finish your lab work during the corresponding lab session.
3. Name the complete document as follows:

LastName\_FirstName\_CS115\_Lab1\_Report.doc

1. Submit the final document to Blackboard Assignments section before the due date. No late submissions will be accepted.
2. ALL lab partners need to submit a report, even if it is the same document.

**Objectives:**

1. (10 points) Demonstrate the ability to break a basic problem down into inputs, process and outputs.
2. (5 points) Demonstrate the ability to design test cases for your problem,
3. (5 points) Demonstrate the ability to select appropriate Java data type for the problem at hand.

**Problem 1:**

Break a basic problem down into **inputs**, **process** and **outputs** and write **pseudocode** (step by step sequence of necessary actions) to solve the problem. **NO Java CODE IS NEEDED**

Answer the following questions for the problems listed below. Populate provided tables (enter as many rows as you find necessary) and pseudocode boxes with your answers. Feel free to add extra tables, boxes, comments, etc. if needed.

INPUTS: What are the inputs?

* What format / data type are they? (integer, real number, single character, string - a sequence of characters)
* Any valid / invalid / illegal / special values? (positive, negative, valid range, etc.)
* How do you get them? (enter manually, ask user, read from file, etc.)

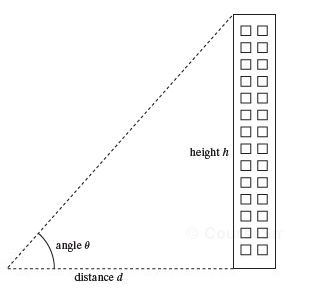
PROCESS: How do you get from inputs to the outputs you want?

* What are the calculation steps?
* To follow these steps, what else do you need? (formulas, etc.)
* Other variables, constants, conversions (besides input and output variables)

OUTPUTS: What are the outputs?

* What format / data type are they in? (integer, floating-point, character, or string)
* Any valid / invalid / illegal / special values? (positive, negative, valid range, etc.)
* How do you output them? (display on screen, save to a file, plot, tabularize, etc.)

1. You can use trigonometry to find the height of a building as shown on the figure below.



Suppose you can measure the angle theta (+/- 3 degrees) between the line of sight to the top of the building and the ground, and you can measure the distance d to the building. Calculate upper and lower estimate for the height of the building. **[3 points]**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Inputs and outputs (use “N/A”, “undefined”, “none”, etc. If necessary)** | | | | | |
| Variable name | Input or  Output? | Data type / format | Constraints | Special cases | Comments |
| height-h | Input | Integer Data  type with feet as the unit | The height-h has to be >0 | If the height-h is <0 the value is undefined | The height can be any positive value. |
|  |  |  |  |  |  |
| angle-t | Input | Integer Data type with degrees as the unit | The angle-t  has to be >0 and <90 | if the angle-t is out of the range of <0 and >90 the value is undefined | The angle has to be between 0 and 90 |
| estimate-u | Output | Integer Data type with feet as the unit of measurement | The estimate-u will be measured using t+3 | If the estimate-u is not equal to t+3 the outcome will be flawed | The estimate-u has to be exactly t+3 |
| estimate-l | Output | Integer Data type with feet as the unit of measurement | The estimate-l will be measured using t-3 | if the estimate-l is not equal to t-3 the outcome will be flawed | The estimate-l has to be exactly t-3 |
| distance-d | Input | Integer Data type with feet as the unit of measurement | The distance-d has to be positive and >0 | If the distance-d <0 then the value is undefined | The distance-d has to be greater than 0 |

|  |
| --- |
| **Pseudocode:** |
| 1. Get the height of the building in feet and the angle of theta in degrees. 2. Use the angle-t of the building and the height-h of the building  to calculate the distance of the ground. 3. Use the angle-h of the building and the height of the building to calculate the distance while estimating the distance from the ground from a higher angle. 4. Use the angle-l of the building and the height of the building to calculate the distance while estimating the distance from the ground from a lower angle. |

1. An outdoor mall is installing raised flower beds. The installer needs to know how many cubic yards of soil in order to fill the flower beds. Each bed is a rectangular box measured in inches (length by width by depth) and there is "count" number of beds. Calculate the total amount of soil to order. **[3 points]**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Inputs and outputs (use “N/A”, “undefined”, “none”, etc. If necessary)** | | | | | |
| Variable name | Input or  Output? | Data type / format | Constraints | Special cases | Comments |
| Length-l | Input | Integer data-type with units as inches | Positive number greater than 0 | If the number is less than 0, then Length should be undefined. | The variable can be close to 0 but not equal to 0 |
| Width-w | Input | Integer data-type with units as inches | Positive number greater than 0 | If the number is less than 0, then Width should be undefined. | The variable can be close to 0 but not equal to 0 |
| Depth-d | Input | Integer data-type with units as inches | Positive number greater than 0 | If the number is less than 0, then Depth should be undefined. | The variable can be close to 0 but not equal to 0 |
| Volume-v | Output | Integer data-type with units as cubic inches | Positive number greater than 0 | Volume can have a small number close to 0 but never equal to 0. | N/A |
| Count-c | Input | Integer data-type | Positive number greater than 0 | Count variable is a natural number[ count>0] | N/A |
| Amount-a | Output | Integer data-type with units as cubic inches | Positive number greater than 0 | Amount can have a small number close to 0 but never equal to 0. | N/A |

|  |
| --- |
| **Pseudocode:** |
| 1. Get the measurements of length, width, and depth of flower-bed in inches. 2. Count the number of flower-beds required in the mall. 3. Calculate the volume of flower-bed by multiplying the length, width and depth and assign the value to volume.[v=l\*w\*d] 4. Multiply the number of flower-beds required with the volume to get the total amount of soil required.[a=c\*v] |

1. If an ice cream cone is 6 inches tall, and its rim has a diameter of 2 inches, write pseudocode to determine the weight of the ice cream that can fit in the cone, assuming that the ice cream above the cone is a perfect hemisphere. You may neglect the thickness of the cone material. Assume that a gallon of ice cream weighs 8 lb and occupies 7.5 cubic feet. **[4 points]**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Inputs and outputs (use “N/A”, “undefined”, “none”, etc. If necessary)** | | | | | |
| Variable name | Input or  Output? | Data type / format | Constraints | Special cases | Comments |
| height | input | The data type for height is integer and is measured in inches | The height is 6 inches | The height cannot be any other integer apart from 6 | If the height of the cone is changed, we will receive a wrong output. |
| diameter | input | The data type for diameter is integer and measured in inches | the diameter is 2 inches | The diameter cannot be any other integer apart from 2 | If the diameter of the cone is changed, we will receive an output that is unintentional |
| volume | output | The data type for diameter is integer and measured in  cubic feet | The volume is 7.5 cubic feet | The volume cannot be any other integer apart from 7.5 | If the height, diameter, and weight change the volume will change. |
| weight | input | The data type for weight is integer and it is measured in pounds. | The weight is 8 pounds | The weight cannot be any other integer apart from 8 | If the weight changes the amount of volume of the ice cream will change. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |
| --- |
| **Pseudocode:** |
| 1. Use the cone formula to figure out the volume of the cone, v = pi r^2 h/3. 2. Plug in the information from the question, such as radius and the height and you willl get the volume of the cone. Answer is 6.4 3. Use the hemisphere formula to figure out the volume of the ice cream over the cone, v = (⅔) pi\*r. Answer is 2.1 4. Add the values together and the total volume of the cone and the ice cream over the cone is 7.5 cubic feet. |

**Problem 2 (5 points):**

Develop a test plan (a set of test cases) for the following problem. The goal of testing is to determine if the solution (a computer program) to your problem:

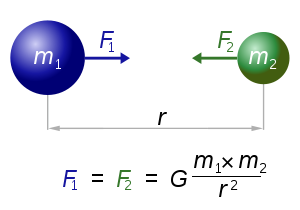
* Behaves correctly / produces correct results when given legal input values,
* Handles illegal input values correctly (for example: preventing division by zero),
* Behaves as planned when inputs are assigned “boundary” values.

In simple words, your test plan, at the very least, should:

* Consider “what could go wrong”,
* Check if your problem solution behaves as expected.

**NO Java CODE IS NEEDED**.

1. Calculating the gravitational force F1 (equal to F2)between two bodies at a distance (see figure below [source: Wikipedia]) **[5 points]**:



|  |  |  |
| --- | --- | --- |
| **Test case name (ex. “negative height”, “typical conditions”, etc.)** | **Input data set for this test case** | **Explain why you chose this test case** |
| Legal Inputs | The distance r, masses m1,m2 are given positive values. | Since mass and distance can’t be negative the Force should be positive as long as considered in scalar quantity. |
| Illegal Inputs | If either the distance, or either of the masses are negative or 0 | The inputs are invalid and the force would be negative |
| Boundary Values | If the either masses are 0, the force can still be calculated. As long as the distance is not 0. | Denominator can’t be 0. As the answer becomes invalid. |
| What could go wrong | If the user puts input data are illegal inputs or the distance input  is 0. G value is defined wrong. | Distance can’t be 0 as then the equation becomes invalid. The constant value can be assigned wrong by a technical mistake. |
| Behaved Correctly | Consider the distance r be 1 metre, mass m1 be 2kg, m2 be 4kg. | While using these inputs the calculated force should be F1=F2=53.384N |
|  |  |  |

**Problem 3 (5 points):**

Which Java data type and why would you use to store following data (if more than one can be chosen, explain your choice). **NO Java CODE IS NEEDED**:

1. US ZIP postal code (basic 5-digit one and NOT the ZIP+4 type) **[1 point]**:

|  |  |
| --- | --- |
| **Java data type(s)** | **Explanation** |
| integer | Since US ZIP code is 5-digit integer can store easily without any loss. |

1. United Kingdom postal code (see: https://en.wikipedia.org/wiki/Postal\_code#United\_Kingdom ) **[1 point]**:

|  |  |
| --- | --- |
| **Java data type(s)** | **Explanation** |
| string | Since UK postal code is alphanumeric, string data type can be used. |

1. Human age **[1 point]**:

|  |  |
| --- | --- |
| **Java data type(s)** | **Explanation** |
| byte | Since human age is atmost 3-digit value. |

1. Final CS 116 grade (A, B, C, D, E or F) **[1 point]**:

|  |  |
| --- | --- |
| **Java data type(s)** | **Explanation** |
| char | Since the grade is a single character. |

1. Earth-Moon distance in yards (420388232.721 yards) **[1 point]**:

|  |  |
| --- | --- |
| **Java data type(s)** | **Explanation** |
| double | Double is a 8-byte datatype which can store 2^64 values. |