ENGG1003
Introduction to Procedural Programming
Lab 3 - Week 4 – 2017
Assessed Lab – Course Value 2.5%

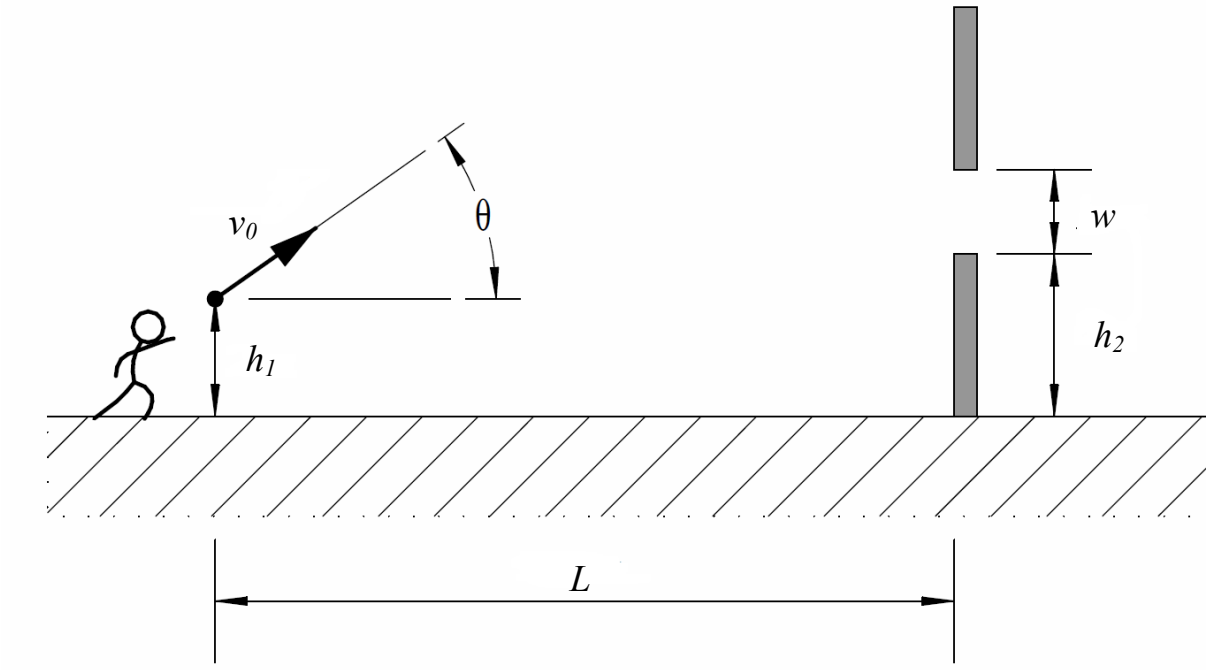
Prerequisites:
It is assumed that you have attended or reviewed lectures, and read Chapters 2, 3 and 4 of the text. It will be of help, if you have tried some of the self-help exercises along the way. **You must attend your lab.**

Lab Objectives:

1. Build and run a simple program using just a C main function (assessed task).
2. Convert this program to use a simple set of user-defined functions (extra task on starting to use functions – not part of the direct assessment for the lab).

Problem Statement:

A person, h_1 (m) tall, attempts to throw a rock through a window in a wall that is L (m) away (see figure below). The window is h_2 (m) off the ground and w (m) high. The rock is thrown with an initial velocity v_0, and an angle of inclination θ (Note that $0^o < \theta < 90^o$).



At some time t after the rock was thrown, the horizontal distance it has travelled, d, is given by

$$d = (v_0 \cos\theta)t$$

and the height of the rock, h, is given by

$$h = -\frac{1}{2}gt^2 + (v_0 \sin\theta)t + h_1$$

where g (m/s^2), is the acceleration due to gravity. The time of flight of the stone to reach the wall, t_w, can be calculated by

$$t_w = \frac{L}{v_0 \cos\theta}$$

and it follows that the height of the stone when it reaches the wall (or window), h_w, is

$$h_w = -\frac{1}{2}gt_w^2 + (v_0 \sin\theta)t_w + h_1$$

If $h_2 < h_w < h_2+w$, then the attempt is successful.

Provided the attempt is successful the total time of flight (t_f) of the stone is obtained by finding the time at which the stone hits the ground (i.e. h=0). This can be obtained by solving following quadratic equation for t_f

$$-\frac{1}{2}gt_f^2 + (v_0 \sin\theta)t_f + h_1 = 0$$

and choosing the appropriate root. Then, the total horizontal distance the rock has travelled, d_f, is obtained according to

$$d_f = (v_0 \cos\theta)t_f$$

Requirements within the Lab:

Develop a C program to do these calculations, just have a main function and incorporate the necessary information, as well as, properly following the documentation requirements presented in lectures.

The program allows the user to enter values of h_1, h_2, w, L, v_0 (velocity), and θ (angle) (Note that you must introduce the user to the application by providing instructions of what information and units are to be input and what will be produced). After entering the data, your application must calculate and show values of t_w and h_w. Then, it must gauge the success of an attempt to throw the stone though the window. If an attempt is successful your program should display "** *Successful* **", and must compute and show the values of t_f and d_w. Otherwise, it should show "** *Unsuccessful* **", and it needs to tell user the distance it missed by. Then the application ends.

Test your program by reporting the following results for the following attempts:

1- A 2.1 m tall basketball player throws a rock through a window in a wall that is 21 m away, at an initial velocity of 15 m/s and an inclination of 35^o. The window is 3m off the ground and 1m high.
2- A 1.75 m tall girl throws a rock through a window in a wall that is 13 m away, at an initial velocity of 12 m/s and an inclination of 45^o. The window is 1m off the ground and 1.5 m high.

Bonus (1 point):
The aim of this part is to find an angle (or angles) for which the projectile will exactly pass through middle of window (i.e. $h_w = h_2 + w / 2$). The user enters all input variables but for the angle of inclination enters minus one (-1) (as a message to the program to apply the bonus calculation). Then, the program must find and display the corresponding angle(s). If there is no solution for the data entered by the user, your program must show an appropriate message.

For the report attempts 1-2 above, find angle(s) of inclination for which the stone will pass through midpoint of window.

Lab Preparation: (part of the assessment – as per *Grading Your Lab* on page 4)

Read through the problem statement.

The analysis part of this lab is based around several equations for which you must find if there are limitations or constraints that must be included in your program. Solve one of the exercises by hand as a way to prepare for the analysis, design and implementation of the program (you must know before attempt any implementation).

Think about the inputs the program will require and what types of variables will be needed to store them, as well as the formats that should be supplied to scanf_s. How can you guide the user into supplying the correct input values (units), and do you need to do any conversions?

Also think about the results to be produced. What variables (and their types) are needed to store the results of the calculations, and what formats are required for each when they are output? What extra information needs to be provided in the output? What other variables might be needed to hold partial results (results not needed to be output)?

Prepare a guide to take with you to the lab. A single A4 handwritten page is likely to be enough. It should contain descriptions of the following:

1. A skeleton of the comment header that you will place at the beginning of your program, including the information shown in the lecture slides,
2. Translate the given formulas into C statements,
3. Pseudocode/IPO chart design of your program. You may sketch out the complete design on paper and take it with you to the lab. Don't worry about making silly mistakes; your main task is to have a *well designed* solution, before you begin with Visual Studio 2015.
4. The types of the input values (with their units) and the variables that will hold these values.
5. The types of the output values (with their units) and the variables that will hold these values.
6. Any unit conversions that will be needed, and any constants you need to define,
7. A list of the formats you intend to use for both the input and the printed results.
8. What input format(s) will be needed to scan in the required input values from the keyboard? How will you need to specify the variable names in the call to scanf_s?
9. A description of the output (values, units, and significant figures of output) including the output format that you will use for each value,
10. Write down some input values that you will be able to use as test data for your program. Choose some values that you can easily verify by hand.

Grading Your Lab:

Indicate to the lab demonstrator when you are ready to have your lab (the working program produced above) graded. Please be patient in waiting, assessing each lab is a non-trivial task. While you are waiting you may proceed with the other problems from the previous computer lab.

The lab demonstrator will first look at the preparation (analysis and documentation) that you brought with you to the lab, and this is to be handed in to the demonstrator.

The lab demonstrator will also assess the program you have produced according to the program layout described in the lecture slides, how easy your program is to read and understand, and on the correctness of the results, and how well the output explains the results that have been produced.

Once you have finished this task, don't forget to call your demonstrator to be assessed.

Cesar Sanin - 2017