

CS 210 - Fundamentals of Programming I

Fall 2019 - Programming Project 2

20 points

Out: September 5, 2019

Due: September 12, 2019 (Thursday)

Reminder: This is a programming project, and work on this assignment should be done individually. Assistance from other students is limited to questions about specific issues as noted in the syllabus.

Problem Statement

As part of an artillery simulation, the military would like to simulate the trajectory of a shell launched by a howitzer gun and determine if the shell will hit its target. The formulas needed for computing the trajectory of any projectile are:

$$t = \frac{d}{v \times \cos(\theta)}$$
$$h = v \times \sin(\theta) \times t - \frac{g \times t^2}{2}$$

where d is distance to the target in feet, v is the projectile velocity in feet/sec, θ is the launch angle of elevation, t is time of flight of the projectile to the target distance in seconds, g is the Earth's gravitational constant in feet/sec², and h is the height of projectile in feet when it reaches the target distance.

Program Specifications

For this assignment, you are given the program specification in the form of an analysis and design for a series of functions that must be implemented to earn full credit for the project. Each function accomplishes one task needed for this program. You should write a function and then test it with a `main` program that ensures that the function works by itself. Once the function is working, move on to the next function, which will require a different `main` function to test it. Once all of the individual functions work, the final `main` function that actually does the simulation is written.

Function: `print_greeting`

Analysis: no parameter or returned objects

Design: This function's task is simply to print out the greeting at the beginning of the program. See the sample run for the exact text format. REMEMBER! Test this function (and all other functions) as you write them.

Function: `get_target_distance`

Function: `get_target_height`

Function: `get_launch_angle`

Function: `get_velocity`

Since we do not, yet, know how get multiple results from one function, we will write a function to ask the user for each of the input values. All four functions share the same analysis and design.

Analysis:

| Objects | Type | Movement | Name |
|-------------|--------|----------|-------|
| input value | double | returned | value |

Design: Each function's task is to prompt the user for one of the input values and return the valued entered. Here are the steps to the accomplish this:

1. Prompt the user for one of the input values
2. Read in the input value
3. Return the input value

See the sample run for the exact text format of each prompt.

Function: degrees_to_radians

Analysis:

| Objects | Type | Movement | Name |
|------------------|--------|----------|---------|
| angle in degrees | double | received | degrees |
| angle in radians | double | returned | radians |

Design: Since the transcendental functions (sin, cos) in the C math library (`<math.h>`) receive their angle argument in radians, but we want the user to enter the launch angle in degrees, the input value will need to be converted to radians. This function's task is to do the conversion. The formula for converting degrees into radians is:

$$radians = \frac{\pi}{180} \times degrees$$

Function: compute_time_of_flight

Analysis:

| Objects | Type | Movement | Name |
|-----------------------------------|--------|----------|----------|
| distance to target | double | received | distance |
| launch angle in degrees | double | received | angle |
| projectile velocity | double | received | velocity |
| time of flight to target distance | double | returned | time |

Design: This function's task is to computes the time of flight for the shell using the formula given above. Note that the launch angle is received **in degrees** (since most people deal with angles in degrees rather than radians).

So the steps of this function are:

1. Compute the launch angle in radians using function `degrees_to_radians`
2. Compute the time of flight
3. Return the time of flight

Function: compute_shell_height

Analysis:

| Objects | Type | Movement | Name |
|--|--------|----------|----------|
| time of flight | double | received | time |
| launch angle in degrees | double | received | angle |
| projectile velocity | double | received | velocity |
| height of shell at the target distance | double | returned | height |

Design: This function's task is to compute the height of the shell at the target's distance using the formula given above. As with the previous function, the launch angle is received in degrees and needs to be converted to radians in the same manner.

Function: print_results

Analysis:

| Objects | Type | Movement | Name |
|---------------------------------|--------|----------|---------------|
| target height | double | received | target_height |
| shell height at target distance | double | received | shell_height |

Design: This function's task is to print the results of the simulation. The steps of this task are:

1. Display the shell's height at the target distance.
2. If the height of the shell is lower than five feet below the target height, print "The shell missed too low"
3. If the height of the shell is higher than five feet above the target height, print "The shell missed too high"
4. Otherwise, print "The shell hit its target"

The displayed value must be formatted with one place after the decimal point. See the sample run for the exact format of output.

Main Program

Analysis:

| Objects | Type | Name |
|--|--------|---------------------|
| distance to target | double | distance_to_target |
| height of target | double | height_of_target |
| launch angle in degrees | double | launch_angle |
| projectile velocity | double | projectile_velocity |
| time of flight to target | double | time_of_flight |
| height of shell at the target distance | double | height_of_shell |

Design: Finally we arrive at the main function of the program, now that we have all the pieces.

1. Print the greeting using the print_greeting function
2. Get the distance to the target using function get_target_distance
3. Get the height of the target using function get_target_height
4. Get the launch angle in degrees using function get_launch_angle
5. Get the projectile velocity using function get_projectile_velocity

6. Compute the time of flight of the projectile using function `compute_time_of_flight`
7. Compute the shell height at the target distance using function `compute_shell_height`
8. Display the results using function `print_results`
9. Return 0 to exit the program

Assignment

Write a C program that implements the shell trajectory simulation. **Your program must follow the specifications given above to earn full credit.** That is, your program must define and use at least the specified functions. (The names of functions and variables do not have to be exactly the same, but the number and types of the parameters must be as specified.)

The output of the program must conform exactly to the following example runs (there are 3 separate runs shown; user input shown in **bold**). Note there is a blank line between the program heading and the user input prompts and there is a blank line before the first line of output. And as usual, there must be a newline after the last line of output.

Coding Notes

- Use 3.1415926 for the value of π and 32.17 for the value of g . These should be defined constants.

Sample Runs

```
SHELL TRAJECTORY CALCULATOR
```

```
Enter the distance to the target (in feet): 6000  
Enter the height of the target (in feet): 1000  
Enter the launch angle of elevation (in degrees): 11.12  
Enter the velocity of the shell (in feet/sec): 1847
```

```
The height of the shell at the target's distance is 1003.0 feet  
The shell hit its target
```

```
SHELL TRAJECTORY CALCULATOR
```

```
Enter the distance to the target (in feet): 5000  
Enter the height of the target (in feet): 1000  
Enter the launch angle of elevation (in degrees): 10.75  
Enter the velocity of the shell (in feet/sec): 1847
```

```
The height of the shell at the target's distance is 827.2 feet  
The shell missed too low
```

```
SHELL TRAJECTORY CALCULATOR
```

```
Enter the distance to the target (in feet): 6000  
Enter the height of the target (in feet): 1000  
Enter the launch angle of elevation (in degrees): 11.5  
Enter the velocity of the shell (in feet/sec): 1847
```

The height of the shell at the target's distance is 1043.9 feet
The shell missed too high

REMINDER: Your program must compile for it to be graded. Submissions that do not compile will be returned for resubmission and assessed a late penalty. Submissions that do not substantially work also will be returned for resubmission and assessed a late penalty.

Follow the program documentation guidelines in the [C Programming Style Guideline](#) handout. As stated in the syllabus, part of the grade on a programming assignment depends on how well you adhere to the guidelines. The grader will look at your code and grade it according to the guidelines. Be sure to run the Source code formatter (Astyle) plugin before you submit your program.

What to Submit

Electronically submit a zipfile containing `main.c` (only) as explained in class and in the handout [Submission Instructions for CS 210](#). The submission system will start accepting assignments no earlier than the evening of Friday, September 6. Reminders: you may submit as many times as needed, and only the last submission will be graded. Assignments must be submitted by 11:59pm to earn full credit.