

CS1XC3 - Computer Science Practice and Experience: Development Basics

Assignment #2 – C Programming Basics

Weight

7% of total course grade

Due date

Sunday February 7th at 11:59pm

Primary Learning Objectives

- Write basic C programs using variables, control structures and functions.

Requirements

Question 1 – Investment calculator – 50 marks

Create a program that asks the user for an initial investment amount, a number of years, a return rate, and an additional contribution per year. The program should then output a neatly formatted table of the investment returns from year one until the final year. All dollar figures in the table should output with 2 digits of precision. The table columns should be: Year, Start Balance, Interest and End Balance. The year column should be 10 characters in length, the others should be 20 characters in length. The data in all columns should be left-aligned. Use a for loop to compute and present the table of data.

The results should be the same as the investment calculator that appears on this page when using the “End Result” calculator (the default calculator that shows when the page first loads) and contributions set to each year: <https://www.calculator.net/investment-calculator.html>. If numbers are somehow off by very small amount like 0.01, that’s fine, don’t worry about that, it’s probably just a small rounding error with the web app. In the example below the additional contribution is added at the start balance at the beginning of each year, in the web app above they add the contribution to the end balance at the end of each year... either is acceptable.

An example of correctly running the program in terms of input and output:

```
Enter initial investment amount: 20000
Enter total years: 10
Enter return rate: 6
Enter additional contribution per year: 1000
```

Year	Start Balance	Interest	End Balance

1	20000.00	1200.00	21200.00
2	22200.00	1332.00	23532.00
3	24532.00	1471.92	26003.92
4	27003.92	1620.24	28624.15
5	29624.15	1777.45	31401.60
6	32401.60	1944.10	34345.70
7	35345.70	2120.74	37466.44
8	38466.44	2307.99	40774.43
9	41774.43	2506.47	44280.89
10	45280.89	2716.85	47997.75

Your program should work either identical or nearly identical this example.

If the user enters a negative number in the case of any of the dollar figures, or 0 or a negative number in the case of the year, print an appropriate error message and ask the user to input the data again. Do not worry about any other kind of input validation (e.g. checking for int vs float or characters). Implement this input validation using a do-while loop for each input.

Here is an example handling issues with input, your program should behave identical or nearly identical to this:

```
Enter initial investment amount: -1
Initial investment cannot be negative.
Enter initial investment amount: 0
Enter total years: -1
Years must be greater than zero.
Enter total years: 0
Years must be greater than zero.
Enter total years: 1
Enter return rate: -1
Rate cannot be negative.
Enter return rate: 0
Enter additional contribution per year: -1
Contribution cannot be negative.
Enter additional contribution per year: 0
```

Year	Start Balance	Interest	End Balance

1	0.00	0.00	0.00

Question 2 – Geometry application - 50 marks

Create a program that presents the user with a menu that has options to:

1. Compute the area of a square
2. Compute the volume of a sphere
3. Compute the surface area of a cube
4. Quit

If the users selects option 4, the program should terminate. If they select options 1, 2, or 3, the program should ask the user to input the appropriate values needed to perform the computation (e.g. side length for a square, radius in the case of a sphere, etc.). The calculation should be carried out and the result should be presented to the user. Assume cm for all lengths. Do not worry about input validation, you can assume the user inputs a number in each case, and you should input and treat the numbers as doubles. Be sure to research how to use scanf to read a double value. Be careful when performing your calculations, make sure that what is output from your program is correct.

Use a do-while to present the menu to the user, and a switch statement to handle the menu choice selection and to carry out the desired computation by calling the appropriate function. Create C functions to handle each computation... each function should take in any necessary measurement as an argument/parameter and return the correct computation. The function should not be responsible for asking for the necessary lengths, do this in your main function. All of these functions can be written in one line, they should not be very long at all. For example, a triangle area function would look like:

```
double triangle_area(double base, double height) { return 0.5 * base * height; }
```

Your program should run identically or near identically to this example:

- 1) Compute the area of a square
- 2) Compute the volume of a sphere
- 3) Compute the surface area of a cube
- 4) Quit

Enter an option: 1

Enter the side length (cm): 10
Area of the square: 100.00 (cm)

- 1) Compute the area of a square
- 2) Compute the volume of a sphere
- 3) Compute the surface area of a cube
- 4) Quit

Enter an option: 2

Enter the radius (cm): 10
Volume of the sphere: 4188.79 (cm)

- 1) Compute the area of a square
- 2) Compute the volume of a sphere
- 3) Compute the surface area of a cube
- 4) Quit

Enter an option: 3

Enter the edge of the cube: 10
Surface area of the cube: 600.00 (cm)

- 1) Compute the area of a square
- 2) Compute the volume of a sphere
- 3) Compute the surface area of a cube
- 4) Quit

Enter an option: 4

Goodbye!

Submission

Save your solution to *Question 1* in a file named **invest.c**, save your solution to *Question 2* in a file named **geo.c**. Put both files in a zip file named **as2.zip** and submit them to the Assignment #2 dropbox on Avenue to Learn.

Marking scheme

We will not be checking for comments and assigning marks for them, but you should sensibly comment your code if you think anything is confusing to the reader!

Component	Description	Marks
Question 1 – Investment Calculator		
for loop	Is a for loop used correctly?	5
Table	Does the table output with the correct formatting?	20
Validation	Is input validation done, and with do-while statements?	15
Computations	Are the correct values output?	10
Question 2 – Geometry Application		
Menu	Is the menu presented to the user, is the option selected handled using a switch?	10
Functions	Are functions used to calculate the relevant values?	15
Formatting	Is the output formatted as required?	5
scanf	Is scanf used to obtain double value input	5
Calculations	Are calculations correct?	15
Total:		100