

Background – “Structs...Scotty, I need more structs!”

The Extra Weekly Assignment for this week is very loosely derived from problem 16.7 in the 4th edition of Deitel and Deitel as well as dialog from nearly every original Star Trek episode ever made. While 16.7 deals with C++ classes (a topic we haven't covered quite yet), this problem uses C structs.

Problem Statement

You are given the following struct definitions for geometric objects: a sphere, cube, and regular tetrahedron

```
struct sphere
{
    float radius;
    float surface_area;

    float volume;
    char name[50];
};

struct cube
{
    float side;
    float face_area;
    float surface_area;
    float volume;
    char name[50];
};

struct regtet
{
    float side;
    float face_area;
    float surface_area;
    float volume;
    char name[50];
};
```

And the following function prototypes:

```
void SetSphere (struct sphere *);
void CalcSphere (struct sphere *);
void DispSphere (struct sphere *);

void SetCube (struct cube *);
void CalcCube (struct cube *);
void DispCube (struct cube *);

void SetRegtet (struct regtet *);
void CalcRegtet (struct regtet *);
void DispRegtet (struct regtet *);
```

Your job is to write a program that will prompt the user to set the radius and name of a sphere, the side and name of a cube, or the side length and name of a tetrahedron. It should be able to calculate the surface area and volume for each shape and the face area for the cube and tetrahedron. The user should also be able to select an option to display the calculated results.

Instructions*Represent*

- Create a flowchart, algorithm, or pseudo code for solving the problem.

Plan

- Create a file named **EWA_09.cpp**
- Outline the steps your program will take by adding comment statements to your file based on the flowchart, algorithm, or pseudo code.

Implement

- In a file called **EWA_09.cpp** create a program that will run indefinitely and prompt the user with the following commands to perform the corresponding tasks
 1. a, A -- Set sphere radius and name
 2. b, B -- Calculate sphere surface area and volume
 3. c, C -- Display sphere radius, surface area, volume, and name
 4. d, D -- Set cube side length and name
 5. e, E -- Calculate cube face area, surface area, and volume
 6. f, F -- Display cube side length, face area, surface area, volume, and name
 7. g, G -- Set regular tetrahedron side length and name

8. h, H -- Calculate tetrahedron face area, surface area, and volume
9. i, I -- Display tetrahedron side length, face area, surface area, volume, and name
10. q, Q -- Quit

- Compile, run, and execute your code

Evaluate

- Perform a hand calculation for **EWA_09.cpp** with several test cases to verify and check your results

Document

- Assemble all of your code, output, and documentation into a single PDF and submit to Carmen according to the DAL.

Include the standard comment and **fprintf()** statements indicating name, seat number, etc.

NOTES:

You may wish to **typedef** your struct definitions to save yourself some typing. You can do this as is demonstrated in Class 34 or by entering something similar to what appears below after your struct definitions, but before your **main()** function.

```
typedef struct sphere Kirk;  
typedef struct cube Scotty;  
typedef struct regtet Spock;
```

If you do **typedef** your struct definitions, you can change your prototypes from those listed above, accordingly as long as each **typedef** occurs before the prototypes in your file.

Get your indefinite loop working first. Add an if-else-if-else or switch-case and get the “quit” task working. Then add **SetSphere()** followed by **CalcSphere()** and finally **DispSphere()**. Once those three are working, it’s copy and paste and mild modification for the rest.