Objectives:

Create a menu that works through several physics calculations

Solve for velocity, acceleration, 4 motion equations, NSL, weight, and momentum

Show the steps to solve (like showing your work)

Display a Main Menu

What would you like to calculate?

1. Velocity

2. Acceleration

3. Motion

4. Force

5. Weight

6. Momentum

E. EXIT

L: Clear

User will input an integer from 1-6. Validate the input.

If the user inputs a string or character for the exit, then validate that and then exit.

If the user types out velocity, accceleration, or whatever string - validate it, make sure the user isn't trying to exit, and then progress through the menu.

Until the user hits exit (Do, While(not pressing exit)

Clear the Screen.

1. Velocity Menu

Velocity

Return to the main menu at any time by typing "E"

Velocity is the change in position divided by the change in time.

What is the change in position?

\*User inputs a double. Validate. \*

What units?

\*User inputs a string. Validate.\*

What is the change in time?

\*User inputs a double. Validate.\*

What units?

\*User inputs a string. Validate\*

units = posUnits + " / " + timeUnits

v=ds/dt (use the delta symbol if possible)

v=\*plug in for the users inputs\*

v= \*the result after doing calculations\* \*units, but after calculations\*

When you are ready to return to the main menu, type 0.

2. Acceleration

This uses the same exact logic as above, but with some different words.

When displaying the units, do this:

m/s\*s or meters / second times second == (m/s)/s or m/s^2

do this one: meters/(s\*s) to be as accurate as possible. We shouldn't assume the user will use conventional units or best practices.

Main Function:

Loop until the user exits

Display the Main Menu, as shown above. (as a function call, make the main menu itself a function.)

Query the User for an option.

Perform input validation on the option

clear the screen

call the menu builder helper function, unless option is motion.

if the option is motion then

Display a new menu going over the 4 available sub-options. (as a function call, making this menu itself a function)

Query the user for an option

perform input validation

compare the option to the mtion calculations available.

clear the screen

call the respective function for whatever motion equation was queried from the user.

On Exit: Say Goodbye.

Menu Builder function

this function will help us to build the menus for our stuff, since we're gonna have a lot of menus.

function menu builder helper (string whatCalculation, string formula, string thing1, string thing2)

Display a menu like this:

whatCalculation

formula

What is this value of the thing1?

\*user inputs. validate double.\*

What units?

\*user inputs. validate string.\*

What is the value of the thing2?

\*user inputs. validate double.\*

What are the units?

\*user inputs. validate string\*

formula

\*\* manipulate the string such that the signs are separated out, and we figure out what operations need to be done. from the string. \*\*

example

v = deltaPos / deltaTime

\*break apart the string\*

"v =" + deltaPos + mathOperation + deltaTime

\*do some comparisons, check what the math operation is.\*

\*call the physics multiplier function to do the math.\*

v = physicsMultiplier(variables)

//Just return a string that does the calculations for us

string physicsMultiplier(bunch of variables))

//the bunch of variables

string units1

string units2

double value1

double value2

bool multiply

if multiply is true then

double result = value 1 \* value 2

make result have only 4 decimal places

return "" + result + " " + units1 + " \* " + units 2

else

double result = value 1 / value 2

make result have only 4 decimal places

return "" + result + " " + units1 / units 2

return "Error. No deets sorry you're your own" (jk provide some explanation if something goes wrong, but nothing should.)

//Motion Functions

function pos(pos0, vel0, acceleration, time)

Display a menu, similar to the one built in the above menu builder function.

Query the user for the initial position, initial velocity, acceleration, and time.

Display the equation.

Display the first step, second step, third step, etc.

Display the result.

Query the user if they want to move back to the main menu.

Clear the screen.

We can actually really easily create a function that takes the given data, and as long as it has enough data, can fill in the gaps and give you any info you need.

Physics and Math are really cool.

However, that is beyond the scope of this project, so instead just create 4 functions - one for each motion equation.

Each other motion equation will basically be the same as the one above. Just display a basic menu - query the user - validate - perform the calculation - display the result - wait for exit (validate input, clear, go back to main menu)