# PROGRAMMING ASSIGNMENT 2

CS1410 - 100 points

## OUTCOMES

After you finish this assignment, you will be able to do the following:
- Define and use structures and enumerations
- Prototype and define functions
- Pass parameters to functions by value and reference
- Format output with **setw**

## DESCRIPTION

Open the **main.cpp** file and perform the following steps. Each step corresponds to a **//TODO** item in **main.cpp**. Leave and **//TODO** comments be and don't delete them.

1. Define a **scoped** enumeration named **ShapeKind** whose possible values are: **CIRCLE**, **SQUARE**, and **RECTANGLE**.
2. Define a **Shape** structure with three members:
   - An enumerated variable named **kind** whose data type is the scoped enumeration **ShapeKind** defined in step 1.
   - Two double members named **length** and **width** representing the length and width of the square or rectangle shapes. For circles, the length and width are the same and represent the diameter of the circle.
3. Define a function with the following prototype for calculating and returning the area of the shape.

   ```
   double area(Shape s);
   ```

4. Define a function with the following prototype for calculating and returning the perimeter of the shape.

   ```
   double perimeter(Shape s);
   ```

5. Define a function with the following prototype:

   ```
   string nameOf(Shape s);
   ```

   Given a shape object, this function returns a string representing its name (**"Circle"**, **"Square"**, and **"Rectangle"**) based on the **kind** member of the **shape**. For example given a structure like this **{ShapeKind::CIRCLE, 10, 10}**, it returns the string "**Circle**".

6. Define a function with the following prototype:

```
void promptAndReadInputFor(Shape& shape);
```

Depending on the **kind** member of the **shape** argument, this function prompts the user (using **cout**) to enter and read the values (using **cin**) of the following:
- The **length** and **width** if the **shape** argument is a rectangle. If the entered **length** and **width** values happen to be the same, the **shape** argument's **kind** should be changed to **ShapeKind::SQUARE**.
- The **length** of a square if the **shape** argument is a square. Assign the entered value to both the **length** and **width** members of the **shape** argument.
- The diameter of a circle if the **shape** argument is a circle. Assign the entered value to both the **length** and **width** members of the **shape** argument.

The following steps are done inside the **main()** function.

7. Define two shape objects one for a square and the other for a rectangle. Initialize their lengths and widths to 0.0.
8. Call the **promptAndReadInputFor()** function on each of the define three shapes.
9. Use **setw** to output a report of the defined shapes in a table-like format similar to the outcome below.

Here is a sample outcome where the rectangle's length and width are different:

```
Enter the diameter of a circle: 5
Enter the length of a square: 6
Enter the length and width of a rectangle: 4 7

        SHAPE      WIDTH      HEIGHT    PERIMETER       AREA
       Circle          5           5       15.708     19.635
       Square          6           6           24         36
    Rectangle          7           4           22         28
```

And here is another sample outcome where the rectangle's length and width are the same:

```
Enter the diameter of a circle: 4
Enter the length of a square: 5
Enter the length and width of a rectangle: 6.4 6.4

        SHAPE      WIDTH      HEIGHT    PERIMETER       AREA
       Circle          4           4      12.5664    12.5664
       Square          5           5           20         25
       Square        6.4         6.4         25.6      40.96
```

Make sure to properly indent your code.

# INSTRUCTIONS

For this assignment, you need to have a GitHub account. If you don't have one already, please sign up for one at https://github.com/.

## Getting the assignment starter code from GitHub:

- Sign in to GitHub.
- Go to the assignment link https://classroom.github.com/a/KO3b8uvu and accept the assignment. This should create a private repository under your GitHub username for this assignment. Click on the given link to open this repository and see the starter code.
- Click on the **Clone or Download** button dropdown and copy the given URL.
- Navigate to your assignments folder (or any folder you want this assignment to be placed in) and open it using Visual Studio code.
- In Visual Studio Code, open a new terminal and then run:

  `wsl`                              (for Windows 10 only)

  `git clone THE_URL_YOU_COPIED`

  This will download the starter code of this assignment from GitHub and create a folder for it with a name like **cs1410-assignment-02-github_username**. This is the folder where your program file(s) (**.cpp** and/or **.h**) should reside.

## Compiling your C++ program:

- Open the assignment folder (**cs1410-assignment-02-github_username)** in Visual Studio Code. Start a new terminal and run:

  `wsl`                      (for Windows 10 only)

- To compile your program run:

  `make`

  This command will call the C++ compiler on your program, compile it, and, if no compilation errors are found, create an executable program named "**run**" for it. If there are compilation errors, read the console error messages and then go back to your source files (.cpp and/or .h) and fix them. Save your changes and run the "**make**" command to compile the program again.

- To run your program, run:

  `./run`

- To clean (remove) old compilation files and start over, run the command:

  `make clean`

  You can now run the "**make**" command to compile your program again and the "**./run**" command to run it.

## Submitting your program to GitHub:

- Make sure to save your changes and commit them to GitHub when you are done. You can do that by running the following commands from inside your assignment folder:

  **`wsl`**                    (for Windows 10 only)

  **`git add .`**

  **`git commit -m 'short commit message goes here'`**

  **`git push`**

  Make sure to do this at least once by the deadline. For your final submission, I recommend using "**Final submission**" for the commit message. **Note that committing changes is not enough; you have to push them to GitHub; otherwise, your changes will stay on your local machine and I will not be able to see your submission.**
- Go to your assignment repository in github.com and make sure your changes are there.
- Click on the **Clone or Download** button dropdown and copy the given URL. Go to Canvas and submit the copied URL. **This URL must be submitted in Canvas after you make your "Final submission" to GitHub.**

## RUBRIC

| CRITERIA | POINTS |
|---|---|
| Scooped enumeration and structure | 20 |
| Functions | 50 |
| Variables | 10 |
| Formatted output with **`setw()`** | 10 |
| Readable, commented, and properly indented code | 10 |
| TOTAL | 100 |