Program 3
Decision Statements
(100 points)

Due: Friday, September 22 on Blackboard by 11:59 PM

Overview

For this assignment, write a program that will demonstrate how income tax is applied towards various levels of income, i.e. what are tax brackets and why making $1 enough into a higher tax bracket doesn't immediately see you paying substantially more in income tax.

Tax brackets as it applies to income tax means that you will pay more taxes for larger incomes. However, each portion of your income is taxed differently. So for example, if the user has a gross income of $100,000, they will pay:

- 10% on the first $11,000 of their income
- 12% on the next $33,725 of their income
- 22% on the next $50,650 of their income
- And finally, 24% on the last $4,625 of their income

So their calculated income tax would be:

```
Income Tax = (0.1 * 11000) + (0.12 * 33725) + (0.22 * 50650) + (0.24 * 4625)
Income Tax = 1100 + 4047 + 11143 + 1110
Income Tax = $17,400
```

Which is *much* different than if they paid 24% income tax on all $100,000 of their income ($24,000).

Basic Program Logic

The basic logic for this program is still similar to what has been done in programs 1 and 2: the user is asked to enter values, calculations are performed, and the results of the calculations are displayed. The new concept is that different calculations will be performed based on values entered by the user.

Ask the user for their gross income. This number may contain a decimal point and must be placed into a `double` variable.

Using multiple decision statements, determine how much income tax they will be expected to pay, using only four tax brackets (rather than the seven the U.S. government uses).

- $0 to $11,000 -- 10%
- $11,000 - $44,725 -- 12%
- $44,725 - $95,375 -- 22%
- $95,375 or more -- 24%

There are at least two approaches to calculating this income tax value:

Approach #1: Cascading if-else statements

By having four different versions of the above Income Tax calculation -- one for each tax bracket used -- you want to execute only **one** of them based on the user's income. For instances where the user's gross income is > $95,375, the only variable in this calculation is how much of their income above $95,375 is taxed at the 24% rate.

The pseudo-code of each if-statement would look something like this:

```
if the user's gross income is less than or equal to $11,000
  Income Tax = only one tax bracket calculation
otherwise, if the user's gross income is greater than $11,000 and less than or equal to $44,725
  Income Tax = the first and second tax bracket calculations
so forth and so on
```

Approach #2: Four if-statements that continuously reduce their gross income

With this approach, you want to build four, single if-statements that ask the same question: is the user's gross income > $0? The reason for this is because you want to apply each tax bracket's calculation to their gross income and then reduce their gross income afterwards. This method would benefit greatly from having a copy of the user's inputted gross income value in a second double variable.

So for example, if the user's gross income is $100,000 as it was above:

```
if their gross income is greater than $11,000 then
  Income Tax = Income Tax + 1100
  gross income = gross income - 11000
otherwise
  Income Tax = Income Tax + (0.1 * gross income)
  gross income = 0
```

If their gross income is still > $0, then there's more taxing that can be done similar to this logic but with the other three brackets. Some of which will be skipped over entirely if the tax bracket isn't necessary, e.g. if the user's gross income is $35,000, there won't be any income taxed at the 22% or 24% bracket.

Program Requirements

1. At the top of the C++ source code, include a documentation box that resembles the one from programs 1 and 2. ***This will be a part of every program that is submitted during the semester and this will be the last reminder in the program write-ups.***
2. Include line documentation. There is no need to document every single line, but logical "chunks" of code should be preceded by a line or two that describes what the "chunk" of code does. ***This will also be a part of every program that is submitted for the remainder of the semester.***
3. The dollar amounts should all be displayed with exactly 2 digits after the decimal point, including zeros.
4. Make sure to test the program with values other than the ones supplied in the sample output.
5. Hand in a copy of the source code (CPP file ONLY) using Blackboard.

Output

A few runs of the program should produce the following results:

Run 1

```
What is your gross income? $100000

Gross Income:       $  100000.00
Income Tax:         $   17400.00
```

Run 2

```
What is your gross income? $75611.34

Gross Income:       $   75611.34
Income Tax:         $   11941.99
```

Run 3 (oh, so this is why the super-rich want to avoid paying taxes...)

```
What is your gross income? $23112456.99

Gross Income:       $23112456.99
Income Tax:         $ 5540389.68
```

Run 4

```
What is your gross income? $44725

Gross Income:       $   44725.00
Income Tax:         $    5147.00
```