InClass 5 - Tax Calculations

Goal

In this programming project, you will be writing an application to calculate the taxes for an individual based on their income and filing status. It will be graded by Web-CAT. **It is the first part of completing the <u>next Homework Assignment</u>**!

Learning Objectives

- o Stepwise Refinement
- o Familiarity with writing complex expressions
- o Using constants
- o Using a header file appropriately
    - o the const definitions go in the header file
    - o WARNING: a header file never contains blocks of code.
- o Do not use functions (I know that isn't an Objective, but you will all notice this up here!)
    - o Read the WHOLE assignment before you start writing code!

Program Features

When your program starts, the user is prompted for a single letter indicating their filing status, and the amount of income the user has made that year. There are 4 statuses: **S**ingle, married filing **J**ointly, **F**iling separate married, and **H**ead of household. **For this assignment, we are only handling the status for Single**, but you must still read in the S for Single. The table below shows the rates for 2016, and the following examples show how the tax is calculated so that you can derive the formula. This table is for the `taxable income`. You must subtract the standard deduction before calculating the tax. The Standard Deduction table is given below.

For example: For someone **S**ingle with a <u>taxable income</u> of $**12,000** (18300 entered for income), the first $9,275 is taxed at 10% and the remaining $2,725 is taxed at 15%. Tax is *1336.25*.

- o ($**9,275** * 0.10) + (($**12,000** - $**9,275**) * 0.15)
- o $927.50 + $408.75
- o $1336.25

2016 U.S. Federal Personal Tax Rates for Just Single

Marginal Tax Rate	Single
10%	Up to $**9,275**
15%	$9,276 - $37,650
25%	$37,651 - $91,150
28%	$91,151 - $190,150
33%	$190,151 - $413,350
35%	$413,351 - $415,050
39.6%	$415,051 or more

Use constants like this:
SINGLE10*PER10 + (SINGLE15-SINGLE10)*PER15 + (SINGLE25-SINGLE15)*PER25
+(190150-91150)*PER28 + (200000-190150)*PER33

```
const int SINGLE10 = 9275;
const int SINGLE15 = 37650;
const int SINGLE25 = 91150;
   …
const double PER10 = .1;
   …
```

You don't use this for this assignment.

Here is the whole table that you will use for the Homework:

Tax Rate	Single	married Filing Jointly	married Filing sep	Head of Household
10%	Up to $9,275	Up to $18,550	Up to 9,275	Up to $13,250
15%	$9,276 - $37,650	$18,551 - $75,300	$9,276—$37,650	$13,251 - $50,400
25%	$37,651 - $91,150	$75,301 - $151,900	$37,651—$75,950	$50,401 - $130,150
28%	$91,151 - $190,150	$151,901 - $231,450	$75,951—$115,725	$130,151—$210,800
33%	$190,151 - $413,350	$231,451 - $413,350	$115,726—$206,675	$210,801—$413,350
35%	$413,351 - $415,050	$413,351 - $466,950	$206,676—$233,475	$413,351—$441,000
39.6%	$415,051 or more	$466,951 or more	$233,476 or more	$441,001 or more

Your program should prompt the user to enter the filing status and income, and from that income value your program should subtract the Standard Deduction and compute the owed tax. You will use S, J, F, and H as input for the status. Here is the <u>Standard Deduction</u> for just Single, which is all you are responsible for this in-class exercise:

Filing Status	2016
Single	$6,300

const int STD_DED = 6300;

For example, from an input of S and 18300, you calculate the tax from the table above for 12000, i.e. 18300 - 6300 = 12000. The tax is $1336.25 as shown above.

Additional examples are given below.

The Standard Deduction should be hard coded using a constant literal. **You must use a <u>const</u> for the standard deduction and <u>all</u> of the constants.**

Additional Requirements and Grading Notes

You must use a `int` for the **income** variable that you use. You will get rounding differences (slightly different values than web-CAT is expecting) in your income tax values if you use float or double. **You must use `double` for the owed tax.**

1) You must use constants for all of the fixed numbers (the percents and the numbers from the above table) in your program. These constants must be used inside your `if` conditions **and** inside your calculations in your code. You may **not** have numbers inside your calculations. You must come up with a reasonable naming scheme, use meaningful identifier names (for all of your constants and variables!), and use proper case conventions in your names. Please remember: variables names start with lower case; classes and structs (which we have not covered) start with capital letters; **constants are in ALL CAPS**.

2) You must put your constants in a separate header file and include it at the top of your program, e.g. if your header file is called consts.h, then the top of your program should look like this:

```
#include <iostream>
#include "consts.h"
```

To submit this assignment, you need to zip together you *.cpp file and you *.h header file. I will cover how to do this in class.

Examples of I/O

```
Please enter your filing status
Enter S for Single filers,
     J for married filing Jointly,
     F for married Filing separately, or
     H for head of household
Status: S
Please enter your income: 123456
Your tax is $25840.43.
```

```
Please enter your filing status
Enter S for Single filers,
     J for married filing Jointly,
     F for married Filing separately, or
     H for head of household
Status: S
Please enter your income: 15576
Your tax is $927.65.
```

All of these examples use the standard deduction shown above, so you subtract 6300 before you calculate the tax owed.

Please notice the punctuation!

Developing the Program

One technique that is helpful in solving problems (whether they are programming problems or otherwise) is **decomposition**: breaking the problem into smaller and manageable pieces. With a program like this, you need a place to collect solutions to those pieces. A natural strategy in object-oriented programming is to create a new class. In this project you are only required to use one class, but must use various methods to solve different pieces of the problem. This technique of writing methods to solve pieces of the problem is known as "Stepwise Refinement."

Here are some helpful hints:

o   Make sure you understand how a single calculation is done by doing several **by hand**!
o   Use "Stepwise Refinement" and decompose the problem (like the comments I have used)
   1. Input
   2. Calculations
   3. Output
o   Write the Calculations code in steps --
   1. Figure out how to do the calculation ONLY for Single, $8,000 and write the code (and test it)
   2. then figure out the calculation for Single, $10,000 (and test it)
   3. then add one if statement to separate the 2 tax brackets (and test it)
   4. then figure out the calculation for Single, $40,000 (and test it)
   5. then add if statements for more of the tax levels/brackets. (and test it)
o   Format your output to 2 decimal places, using fixed and setprecision()
o   Create a variable for each piece of data you need, then prompt the user for that data and store it in your variables.
o   You may not use functions for this assignment.

Submit your program to Web-CAT

An extensive set of tests will be run against your program, about 12. For example all of your prompts will be tested. Also each and every boundary between each tax bracket, e.g. 9275 and 9276 for Single. These 2 tests show that you have the calculation correct for 9275 at 10% and that the tax for 1 more dollar is 0.15 more, i.e. 927.65. This type of methodical testing is call Boundary Value Analysis. It will test all of your if statement Boolean expressions as well as your calculations.