CS 1400-03 Introduction to Programming and Problem Solving
Project #2
(Due: 11:59 PM, Monday, 2/22/2021)

Write a program that takes as input the marital status ("`single`" or "`married`", case insensitive) and the taxable income (`double`), and computes taxes for the following schedule.

If your status is Single and if the taxable income is over	But not over	The tax is	Of the amount over
$0	$8,000	10%	$0
$8,000	$32,000	$800+15%	$8,000
$32,000		$4,400+25%	$32,000
If your status is Married and if the taxable income is over	But not over	The tax is	Of the amount over
$0	$16,000	10%	$0
$16,000	$64,000	$1,600+15%	$16,000
$64,000		$8,800+25%	$64,000

For example, if David is single and his taxable income is $10,000, then his income tax will be

$800 + 15% of ($10,000 - $8,000) = $1,100

The following are examples of the required I/O behavior, where the user's input is shown in bold.

```
fcsang@fluffy ~/cs1400/project $ java Tax
Enter your marital status (single or married): sinGLE
Enter your taxable income: 10000
Your income tax is $1,100.00

fcsang@fluffy ~/cs1400/project $ java Tax
Enter your marital status (single or married): MARRied
Enter your taxable income: 64001
Your income tax is $8,800.25

fcsang@fluffy ~/cs1400/project $ java Tax
Enter your marital status (single or married): unknown
-- illegal marital status --

fcsang@fluffy ~/cs1400/project $ java Tax
Enter your marital status (single or married): single
Enter your taxable income: -100
-- illegal income --
```

The program should use a constant (declared with the modifier `final`) to represent each base tax (highlighted in yellow) and tax rate (highlighted in red). If an invalid value of marital status or invalid taxable income is input, the program should display an error message and stop. Use format specifiers to display taxes with $ in the front, comma grouping separator, and exactly two digits after decimal point with trailing zeros displayed.

Tips:
Break this problem down into smaller pieces. This is always good practice when tackling a larger problem. Break it up into pieces that you can test individually and work on one piece at a time. You might try writing the pseudo-code for each piece.

First, see if you can get a marital status and ensure that it is valid. Test out only this piece of functionality before continuing. Make sure you test both the good and bad cases. For example, try entering single, SINGLE, sinGLE, singled, married, MARRIED, and marrie.

Next, see if you can get a taxable income and ensure that it is valid. Test this piece too.

Finally, use the now-valid marital status and taxable income to compute taxes. If you tested the earlier pieces, you will now know that any bugs are due to a problem here.

Submission:
In your `cs1400/project` directory, create a Java Source Code file named `Tax.java`. Your Java program must begin with the comments below and follow the naming and coding conventions posted on Blackboard.

```
// your name
// CS1400, section 03
// Project 2 – Income Tax program
// date
```

Generate a `script` file `pj2.txt` with appropriate time stamps and the following steps visible:

1. a `pwd` to show the current working directory
2. a `ls -l` to show in long format the files in your `cs1400/project` directory
3. a `cat` to display `Tax.java`
4. compile `Tax.java`
5. run `Tax` many times using all the following test cases in order to show the correctness of your program

Normal Cases	(single, 1000), (SINGLE, 10000), (sinGLE, 40000), (married, 1000), (MARRIED, 20000), (MARRied, 80000)
Boundary Cases	(single, 0), (single, 1), (single, 7999), (single, 8000), (single, 8001), (single, 31999), (single, 32000), (single, 32001), (married, 0), (married, 1), (married, 15999), (married, 16000), (married, 16001), (married, 63999), (married, 64000), (married, 64001)
Abnormal Cases	Invalid marital status: unknown, s, sing, singled, m, marrie Invalid taxable income: -1, -100

Submit the file `pj2.txt` on Gradescope.