/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Workshop #2 Course:JAC444 -Semester 4

Last Name:Chan

First Name:Daryan

ID:113973192

Section:<section name>

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature Daryan Chan

Date: 6/7/2020

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Detailed Document

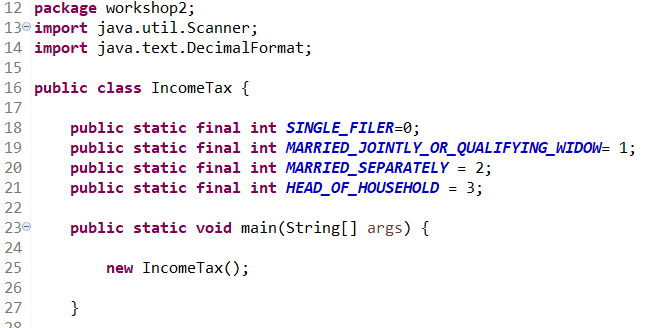


Figure First part

The first part I coded were the constants. These were the constants I had to follow from the instructions. I made them public so that my functions later can access them and made them final so that it can’t be modified. The next part was coding the main and is creating a new IncomeTax object, which essentially invokes the constructor.

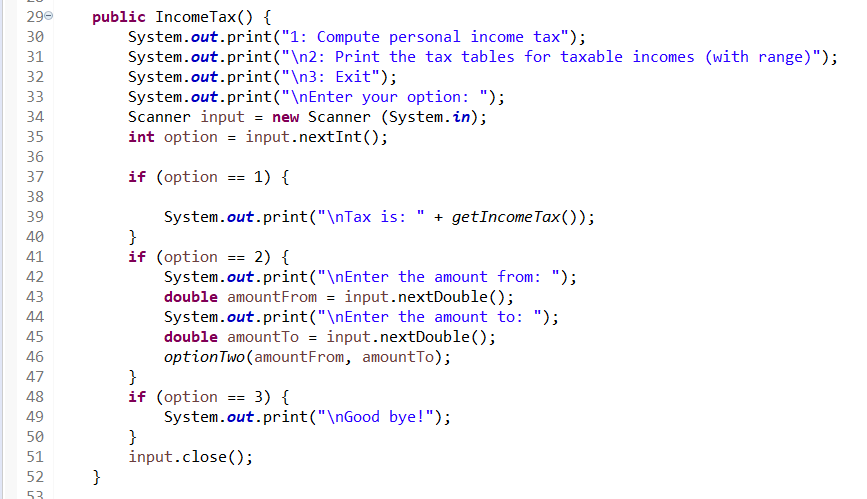


Figure IncomeTax constructor

The constructor invoked here will display command prompts to the user. The program will act depending on the user’s desired function. There are 3 options the user can choose from and the console will display further instructions based on those options. If option 1 is called, it will display what the tax is and calls another function called getIncomeTax. Option 2 will require user to enter a range and call another function called optionTwo. Third option will say good bye and end the program.

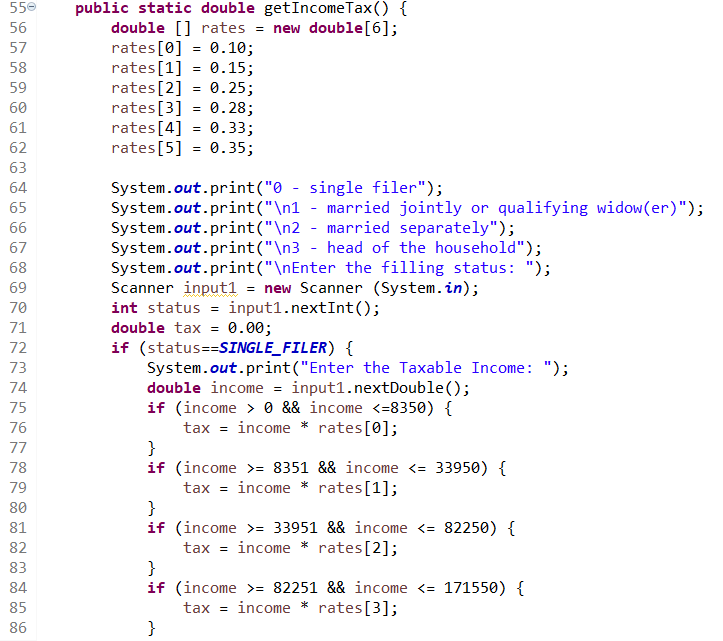
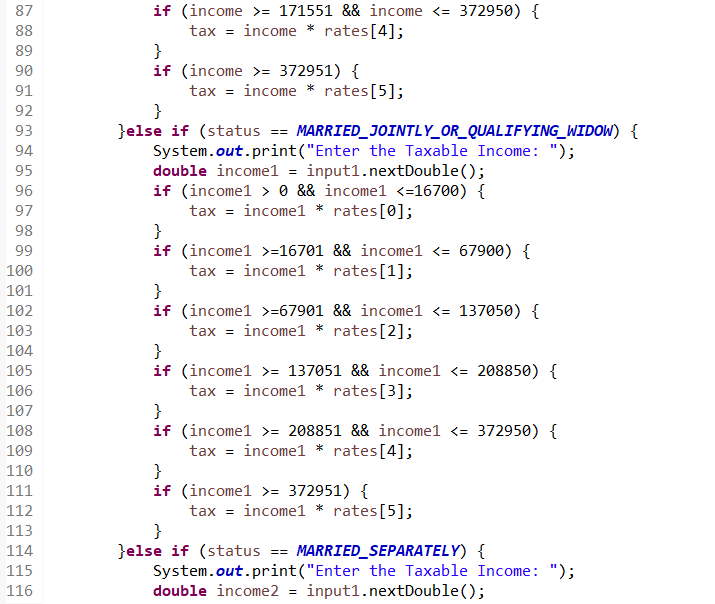
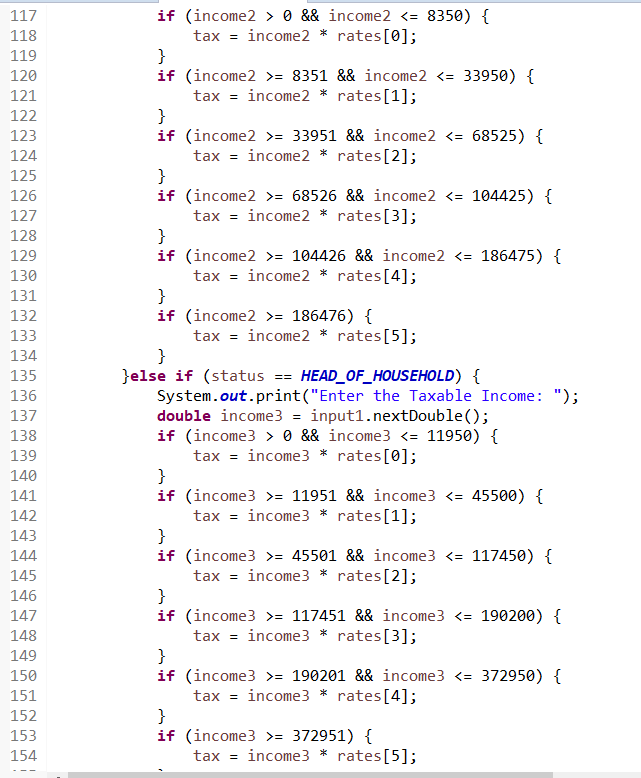
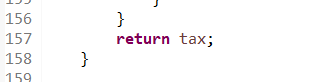


Figure getIncomeTax function

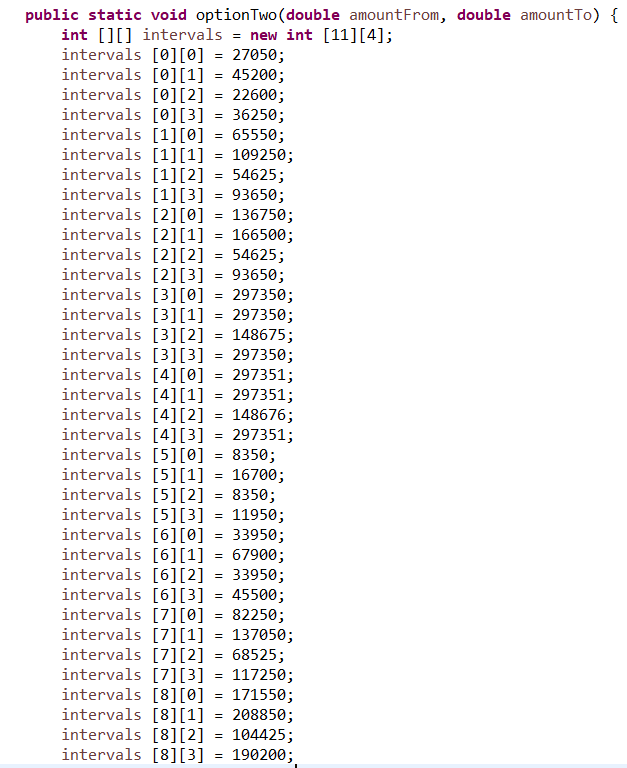
This function begins with me coding an array to store the tax rates. When this function is called, it will display further command prompts and the user has to choose which category the user will file their tax under. When the option is entered, it will be stored in the variable status. The status will then go through multiple if statements to see if it matches with the constants. Once the status is matched with a constant, it will calculate the tax by bringing up the correct tax rate. The rest of the logic is displayed below.







Once the calculation is done, the function will return the tax back.



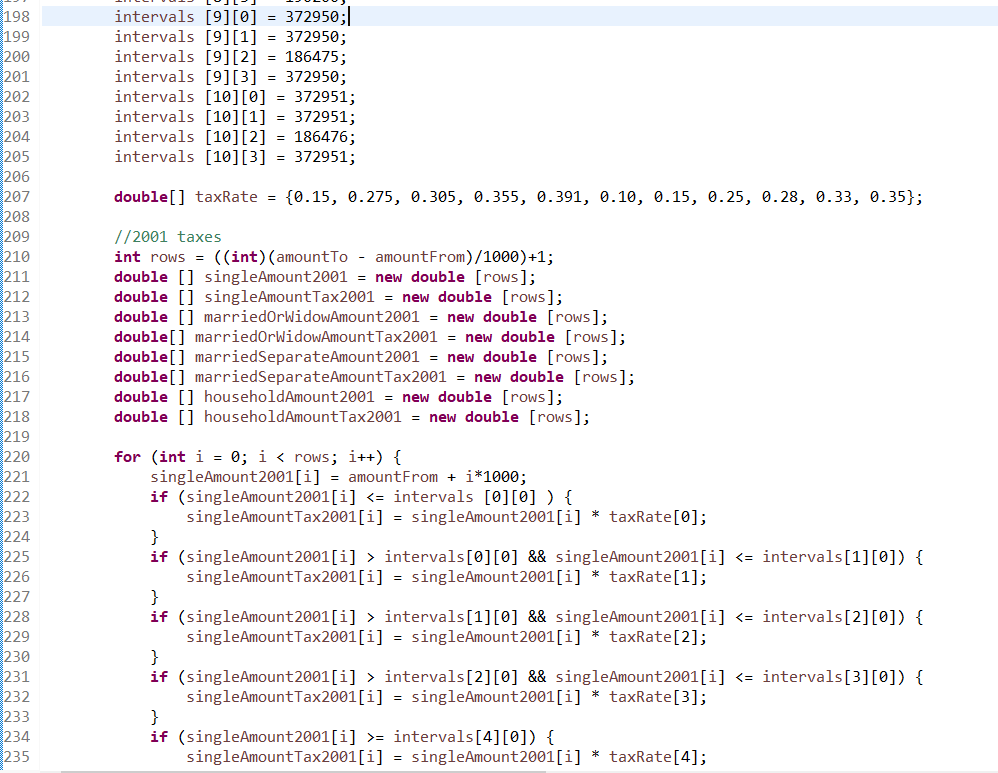
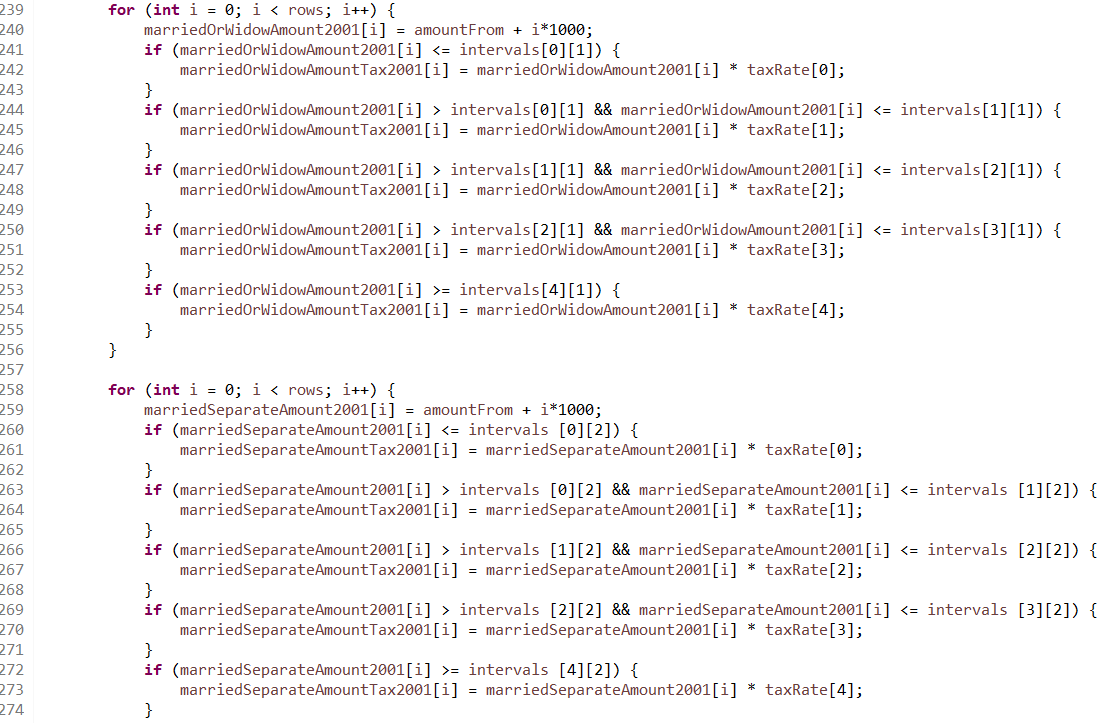
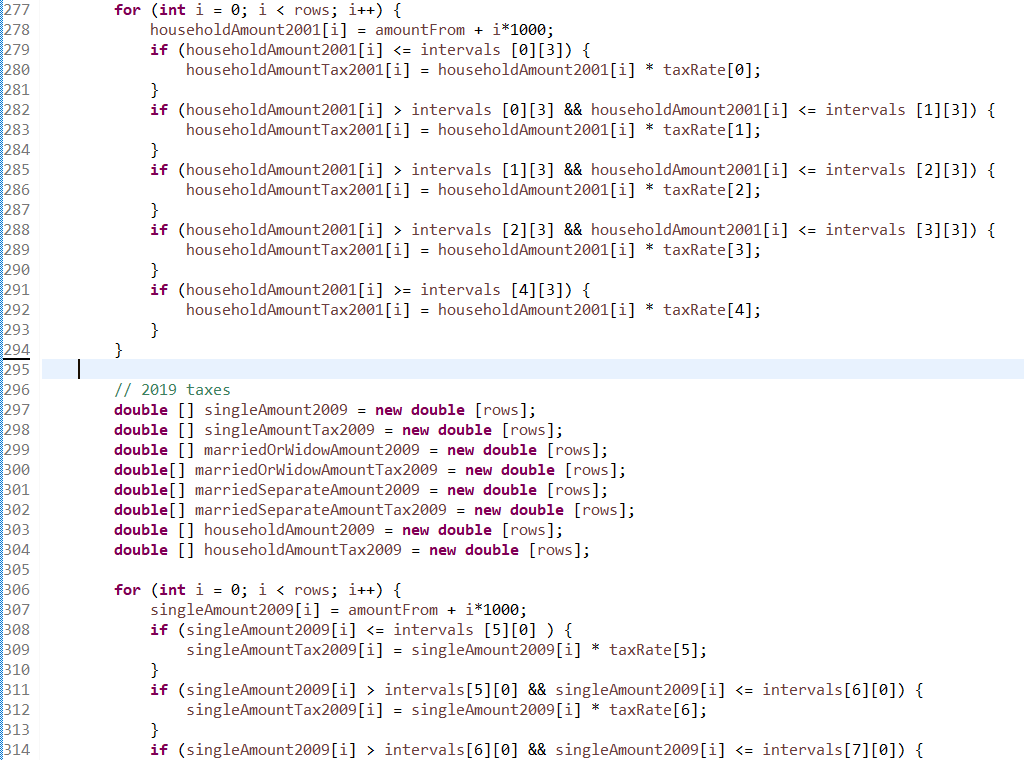
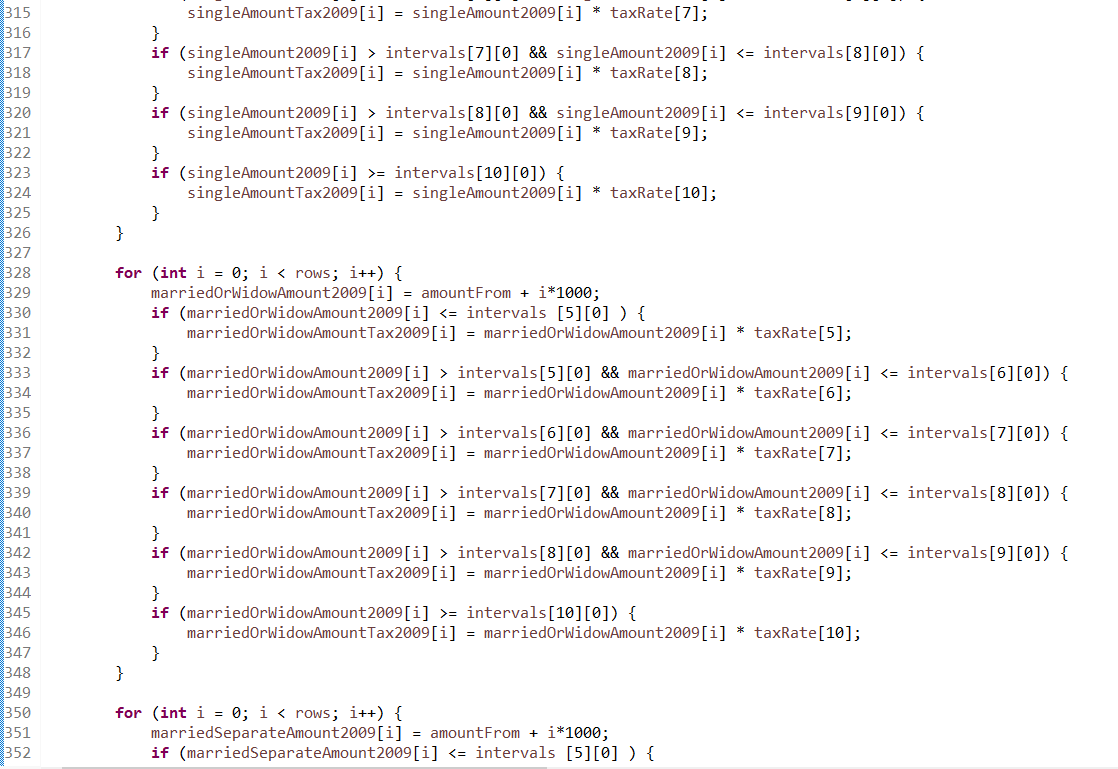


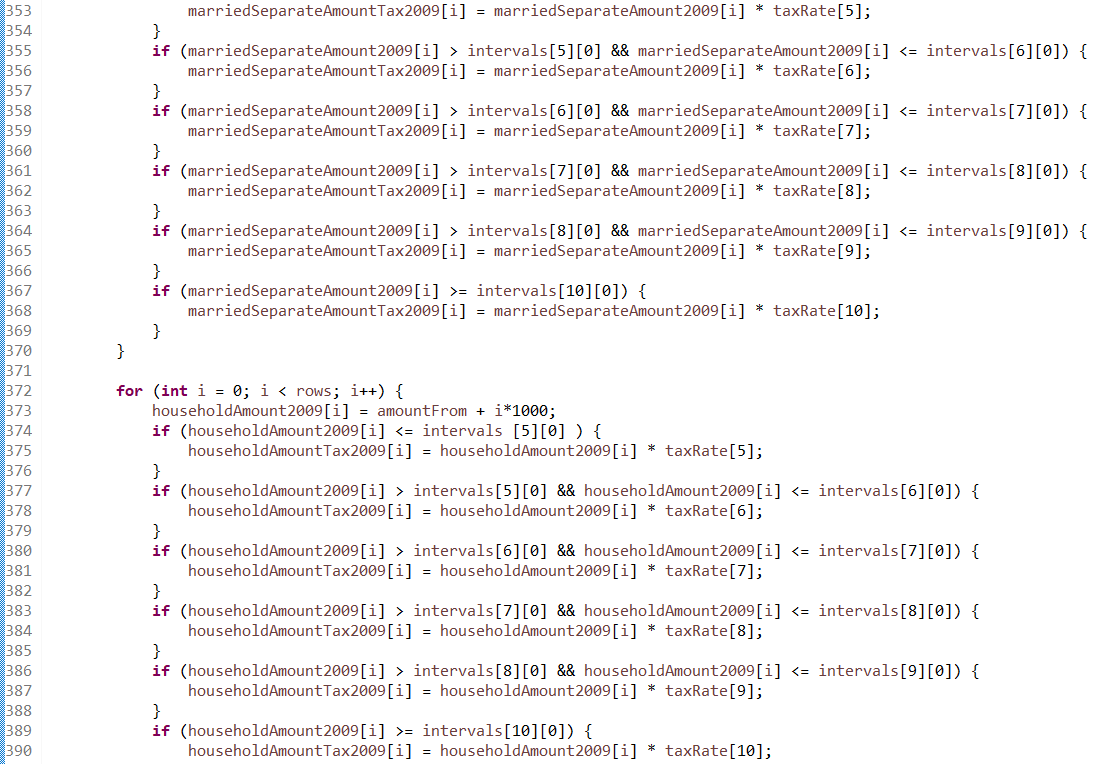
Figure optionTwo function

This function receives 2 parameters, the amountTo and amountFrom. Those two will represent the range that the user has entered. The first thing I coded for this function is the 2d array, which contains the intervals for 2001 and 2009. After, I made an array containing the 2001 and 2009 tax rates. Rows will be calculated based on the range the user entered and I casted it to int in case the user enters numbers with decimals. So I will need to convert it from a double to an integer. Next, I made several variables to help calculate the taxes based on the category and year. The arrays will be set to the number of rows there are based on the earlier calculations. Then in a for loop, it calculates the tax amount for each category and will continue until the end of the row. The screenshots below is basically a continuation of the calculations.









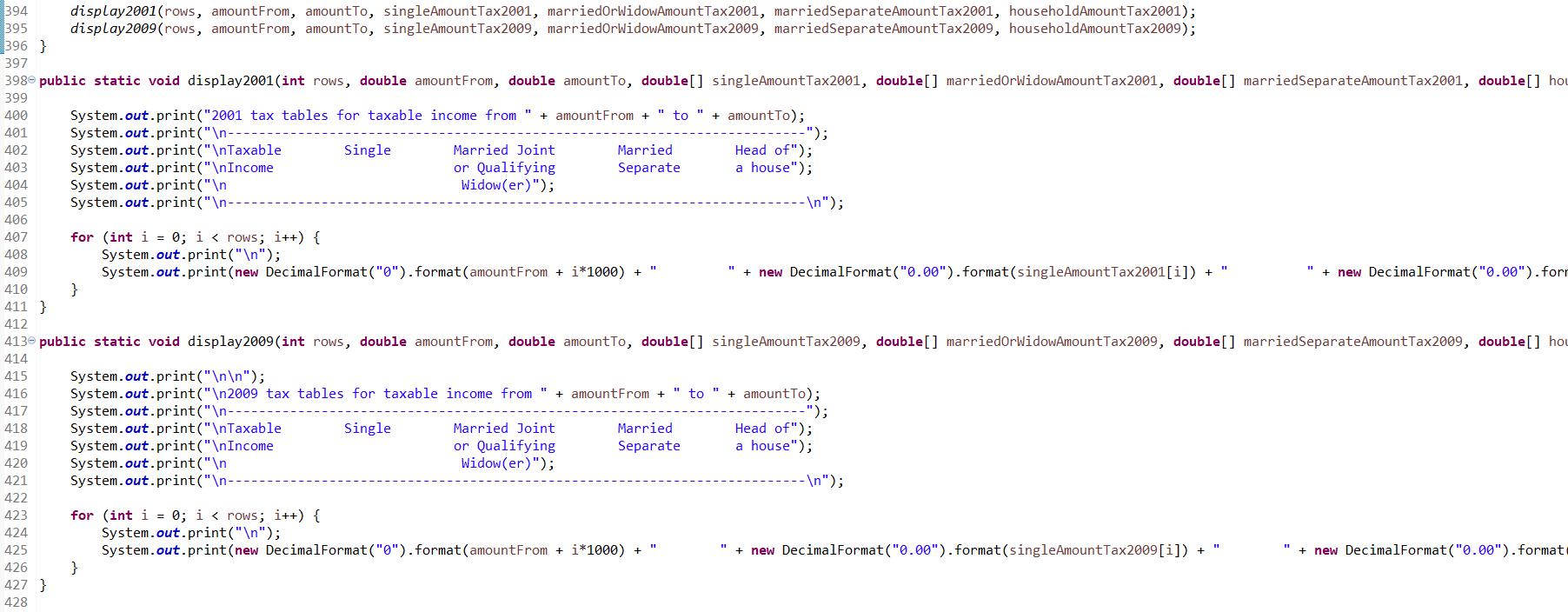
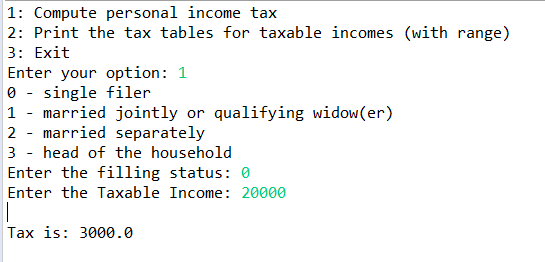


Figure displaying tax table

Once the calculations are calculated, it will call display2001 and display2009 functions. It will pass the rows, amountFrom, amountTo, the tax arrays for each category and tax years and will display them all in a table, as shown in the instructions samples. To make sure the decimal points are formatted correctly, I used the new DecimalFormat().format() method.

Running the program using option 1 and under single filer status



Running the program under option 2 and with ranges

