Student Name: Ankan Basu\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class and Section CSC201 M6\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Total Points (50 points) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Project: Computing Income Tax**

CSC 201 – Computer Science I

New River Community College

Problem Description:

The United States federal personal income tax is calculated based on filing status and taxable income. There are three filing statuses: single filers, married joint filers, and head of household filers. The tax rates vary every year. Table below shows the expected rates for the year 2013. If you are, say, single with a taxable income of $10,000, the first $8,925 is taxed at 10% and the other $1,075 is taxed at 15%. Your tax will be $1,053.75.

|  |  |  |  |
| --- | --- | --- | --- |
| Rate | Single Filers | Married Joint Filers | Head of Household Filers |
| 10% | $0 to $8,925 | $0 to $17,850 | $0 to $12,750 |
| $15% | $8,925 to $36,250 | $17,850 to $72,500 | $12,750 to $48,600 |
| 25% | $36,250 to $87,850 | $72,500 to $146,400 | $48,600 to $125,450 |
| 28% | $87,850 to $183,250 | $146,400 to $223,050 | $125,450 to $203,150 |
| 33% | $183,250 to $398,350 | $223,050 to $398,350 | $203,150 to $398,350 |
| 35% | $398,350 to $400,000 | $398,350 to $450,000 | $398,350 to $425,000 |
| 39.6% | $400,000 and up | $450,000 and up | $425,000 and up |

You are to write a program to compute personal income tax. Your program should prompt the user to enter the filing status (Enter 0 for single filers, 1 for married filing jointly, and 2 for head of household filers) and the taxable income. Then the program should compute the tax and display the tax.

Important note: Use constants for tax rates and variables for tax brackets

**Here is a sample run:**

What is your filing status?

Enter 0 for Single Filers;

1 for Married joint Filers;

2 for Head of Household Filers.

0

Please enter your taxable income:

10000

Your Tax is: $1053.75

Analysis:

(Describe the purpose, processing, input and output in your own words.)

**Purpose:** Calculate income tax based on filing status.

**Processing:** I have gone through several stages of processing the data, input structure and logic required for the program to work. Development of the basic program structure took the least amount of time. Modifying the program from a procedural style to an object oriented style was challenging. However, I have spent the most amount of time to implement error handling for the scanner class. The “TRY-CATCH” syntax was very confusing in the beginning, and continuing the cycle of user inputs until the right condition is met took the longest time to code.

The JAR file I have created has four versions of the program. All working and all have user input validation. The version 4 of the program is in its most advanced stage where I have tried to implement OOP concept for income tax calculation.

|  |  |  |
| --- | --- | --- |
| **Code Version** | **Processing / logic / input and outputs** | **Code Version in JAR file** |
| First Development Stage  (Version 0) | In this stage, I have used   * 7 constant for tax rates * 12 variables to store upper and lower limits of a single family tax brackets * 12 variables to store upper and lower limits of a joint family tax brackets * 12 variables to store upper and lower limits of a head of the family tax brackets   Next, I have used switch along with if else conditions to calculate tax. There are 3 cases, one for each filing status.  I thought that I was using too many variables and decided to moed to the next version of code. This version not even saved. | Not saved  Constant=7  Variables=36  Hours coding: 2 |
| **Version 1** | In this stage, instead of using variables for both upper and lower limits, I have used variables only the upper limits for each tax brackets; this step cut down the number of required variable tremendously.  I am still using 3 cases to calculate tax using switch and if-else conditions.   * 7 constant for tax rates * 6 variables to store upper limits of a single family tax brackets * 6 variables to store upper limits of a joint family tax brackets * 6 variables to store upper limits of a head of the family tax brackets   Code saved in the JAR file. | **IncomeTaxCalculator**  Constant=7  Variables=18  Hours coding: 1 |
| **Version 2** | In the previous version, I was able to reduce the number of variables by a great number, but I was not happy with the true LOGIC of tax calculation.  I felt, that I am doing the same exact LOGICAL calculation for each of the tax groups.  So, I decided to reduce the number of variables even more.  In this version, I have used   * 7 constant for tax rates * Only 6 variables to store upper limits of tax brackets   Now, I have used SWITCH statements to assign values to the 6 variable based on the filing status. This also reduces the number of lines required for the program.  Now, I only need to run the true logic only **once!**  Code saved in the JAR file. | **IncomeTaxCalculator2**  Constant=7  Variables=6  Hours coding: 2 |
| **Version 3** | This is a cleaning up version to update the previous version with methods.  In this version, the main() method has the same number of constants and variables, but it uses function calls to properly assign filing status and calculate income tax.  Code saved in the JAR file. | **IncomeTaxCalculator3**  Constant=7  Variables=6  Hours coding: 2 |
| **Version 4** | In the last version of the code, the main() method was taken out from the version 3 code to make it a class. Then an object was created usong the **IncomeTaxCalculator4\_OOP** class to perform required tax calculations. | Constant=7  Variables=6  Hours coding: 4 |

Note: The hours coding reported does not include time I spent implementing user-input validation. It took me more than 10 hours over 4 day period to finally implement validation that seems to be working.

**Design:**

(Describe the major steps for solving the problem.)

I have described the major steps in the table above; in short, it includes:

* Defining required constants and variables
* Defining logic
* Getting user inputs
* Using conditionals to calculate tax
* Presenting output to the console.

**Testing: (Describe how you test this program)**

Testing of the outputs are conducted using hand calculator and using excel spreadsheet.

A few example outputs are presented below.

|  |
| --- |
| Example output from **IncomeTaxCalculator4\_OOP\_Main.main**({ }); |
| IncomeTaxCalculator4\_OOP\_Main.main({ });  What is your filing status?  Enter 0 for Single Filers;  1 for Married joint Filers;  2 for Head of Household Filers.  1  Please enter your taxable income:  10000  Your tax is: $1000.0  IncomeTaxCalculator4\_OOP\_Main.main({ });  What is your filing status?  Enter 0 for Single Filers;  1 for Married joint Filers;  2 for Head of Household Filers.  2  Please enter your taxable income:  10000  Your tax is: $1000.0  IncomeTaxCalculator4\_OOP\_Main.main({ });  What is your filing status?  Enter 0 for Single Filers;  1 for Married joint Filers;  2 for Head of Household Filers.  0  Please enter your taxable income:  500000  Your tax is: $155763.75  IncomeTaxCalculator4\_OOP\_Main.main({ });  What is your filing status?  Enter 0 for Single Filers;  1 for Married joint Filers;  2 for Head of Household Filers.  1  Please enter your taxable income:  500000  Your tax is: $145646.0  IncomeTaxCalculator4\_OOP\_Main.main({ });  What is your filing status?  Enter 0 for Single Filers;  1 for Married joint Filers;  2 for Head of Household Filers.  2  Please enter your taxable income:  500000  Your tax is: $151064.5 |
| Example outputs from IncomeTaxCalculator.main({ }); |
| 1 for Married joint Filers;  2 for Head of Household Filers.  -10  ---------------------------------  Please correct your filing status  Enter 0 for Single Filers;  1 for Married joint Filers;  2 for Head of Household Filers.  1  Please enter your taxable income:  sd  ---------------------------------  Please correct your taxable income  Please enter your taxable income:  -100  ---------------------------------  Please correct your filing status, negetive income not allowed  Please enter your taxable income:  500000  Your tax is: $145646.0  IncomeTaxCalculator.main({ });  What is your filing status?  Enter 0 for Single Filers;  1 for Married joint Filers;  2 for Head of Household Filers.  2  Please enter your taxable income:  500000  Your tax is: $151064.5  IncomeTaxCalculator.main({ });  What is your filing status?  Enter 0 for Single Filers;  1 for Married joint Filers;  2 for Head of Household Filers.  2  Please enter your taxable income:  500000  Your tax is: $151064.5  IncomeTaxCalculator.main({ });  What is your filing status?  Enter 0 for Single Filers;  1 for Married joint Filers;  2 for Head of Household Filers.  1  Please enter your taxable income:  500000  Your tax is: $145646.0 |
| Example output from the submitted “Executable JAR file”  Please note that you open the JAR package and run other version of the codes too. JAR file uses the VERSION 4 of the code with OOP implementation. |
|  |

How to submit your assignment

1. Login Blackboard
2. Click on Assignments on the left
3. Click on Week 4 Work folder
4. Read the instruction there and submit the following items:

* Your jar file with source code. The jar file without the source code will not be graded. Please use the steps given on Project 1 Instructions to create your jar file. Rename your jar file as YourName\_Project2. Suppose your name is Susan Boyd, you should rename your jar file as SusanBoyd\_Project2. Files with wrong name will not be graded.
* This document with answers for analysis, design and testing. Rename this document as Project2\_Yourname. Suppose your name is Susan Boyd, you should rename this document as Project2\_SusanBoyd. Files with wrong name will not be graded. This document is worth 10 points and the comments in your program is worth 10 points. Working code is worth 30 points.