COSC2135 - Programming 1
Study Period 1, 2014
Assignment 1

NOTE: This assignment is to be undertaken individually – no group work is permitted.

Background information

For this assignment you need to write a console application in the Java programming language which implements the functionality of a basic tax calculation program.

Taxation is calculated for a specified financial year based upon a variety of factors:

- Assessable gross income for the previous 12 months
- any bank interest that has been accrued for the previous 12 months
- any additional pre-tax superannuation contributions
- any claimable deductions for things such as work expenses
- the medicare levy for taxpayers who do not have private health insurance

Pre--tax superannuation contributions and claimable deductions are factored in as "offsets" from assessable income in order to determine taxable income for the taxpayer, after which the tax payable amount can be calculated. Any taxpayer who does not have private health insurance is required to pay an extra **1.5%** of their taxable income (on top of any tax they are already required to pay).

Taxation rates that are applied to taxable income (minus super contributions and deductions) are structured in a "tiered" system in which taxpayers who have higher levels of taxable income are taxed at higher rates.

The requirements for this assignment are discussed in three "stages", where each "stage" addresses new concepts by adding more detail to the problem description from the previous stage(s).

You are required to write a program for each of the stages described below.

Disclaimer:

While the scenarios described in the different "stages" of this assignment are based upon current Australian taxation guidelines, the scenarios themselves are intended to represent simplified version of those taxation guidelines and thus they are not meant to be a 100% accurate simulation of the relevant taxation rules and regulations.

Stage 1 – Basic Income Tax Calculator (30 marks)

To begin with the tax calculator needs to gather some basic information from the user about the tax payer: name, tax file number and the financial year for which tax is being calculated.

The basic calculation for the tax that is payable by a tax payer requires the following taxation items to be entered by the user for the specified financial year:

- Assessable gross income earned for the previous 12 months.
- The amount of bank interest that has been accrued in the previous 12 month period.
- The pre-tax superannuation contribution made by the taxpayer in the previous 12 month period.
- The amount of claimable deductions the taxpayer has submitted for the previous 12 month period.

The program should prompt the user to enter these values and then read them into the program from the keyboard and store them in appropriately-typed variables for later use.

Once all required values have been entered you will need to calculate the total assessable income amount (**gross income + bank interest**) and the total amount for tax offsets (**superannuation contributions + claimable deductions**).

From there you can calculate the taxable income for the taxpayer and the amount of tax that they are required to pay as follows:

```
taxable income = assessable income - total offsets

tax payable = taxable income * tax rate
```

Note: As we are just getting started we will assume a basic tax rate of 15% applies to all tax payers regardless of how high their taxable income is for now – this will change in stage 2.

After the calculation of the final tax payable amount has been completed the program should display a summary of all of the data to the screen in a neat, tabulated format (**you must use field width modifiers appropriately in `printf()` to do so**) as shown in the sample execution runs below/on the next page.

Note: You do not have to worry about handling situations where the tax calculation will produce a negative tax payable amount – when the markers test the program they will instructed to avoid input combinations that will produce a negative tax payable amount as a result.

Testing basic wage tax calculation

```
*** Basic Income Tax Calculator ***

Enter Name: Bob The Builder

Enter Tax File Number (TFN): 123 456 789

Enter financial year: 2013-2014

Enter assessable income for period 2013-2014: 10000

Enter bank interest accrued: 0

Enter pre-tax superannuation contribution for period 2013-2014: 0

Enter claimable deduction(s) for period 2013-2014: 0

       *** Final Tax Statement ***

Name:                    Bob The Builder
Tax File Number:         123 456 789
Financial Year:          2013-2014

Assessable Income:       $    10000.00
Minus Tax Offsets:       $       -0.00
Taxable Income:          $    10000.00
Tax Payable:             $     1500.00
```

Note: Assessable income figure corrected 13/03/14

🟦 = user input

🟥 = program variable output

Testing (wage + bank interest) tax calculation

```
*** Basic Income Tax Calculator ***

Enter Name: Bob The Builder

Enter Tax File Number (TFN): 123 456 789

Enter financial year: 2013-2014

Enter assessable income for period 2013-2014: 10000

Enter bank interest accrued: 5000

Enter pre-tax superannuation contribution for period 2013-2014: 0

Enter claimable deduction(s) for period 2013-2014: 0

       *** Final Tax Statement ***

Name:                    Bob The Builder
Tax File Number:         123 456 789
Financial Year:          2013-2014

Assessable Income:       $    15000.00
Minus Tax Offsets:       $       -0.00
Taxable Income:          $    15000.00
Tax Payable:             $     2250.00
```

Note: Tax Payable figure corrected 13/03/14

🟦 = user input

🟥 = program variable output

Testing (wage + bank interest) – super contribution tax calculation

```
*** Basic Income Tax Calculator ***

Enter Name: Bob The Builder

Enter Tax File Number (TFN): 123 456 789

Enter financial year: 2013-2014

Enter assessable income for period 2013-2014: 10000

Enter bank interest accrued: 5000

Enter pre-tax superannuation contribution for period 2013-2014: 3000

Enter claimable deduction(s) for period 2013-2014: 0

       *** Final Tax Statement ***

Name:                        Bob The Builder
Tax File Number:             123 456 789
Financial Year:              2013-2014

Assessable Income:           $     15000.00
Minus Tax Offsets:           $     -3000.00
Taxable Income:              $     12000.00
Tax Payable:                 $      1800.00
```

Note: Financial year prompt was missing - corrected 13/03/14

🟦 = user input

🟥 = program variable output

Testing (wage + bank interest) - (super contribution + deductions) tax calculation

```
*** Basic Income Tax Calculator ***

Enter Name: Bob The Builder

Enter Tax File Number (TFN): 123 456 789

Enter financial year: 2013-2014

Enter assessable income for period 2013-2014: 10000

Enter bank interest accrued: 5000

Enter pre-tax superannuation contribution for period 2013-2014: 3000

Enter claimable deduction(s) for period 2013-2014: 2000

       *** Final Tax Statement ***

Name:                        Bob The Builder
Tax File Number:             123 456 789
Financial Year:              2013-2014

Assessable Income:           $     15000.00
Minus Tax Offsets:           $     -5000.00
Taxable Income:              $     10000.00
Tax Payable:                 $      1500.00
```

🟦 = user input

🟥 = program variable output

<u>Stage 2 – Enhanced Income Tax Calculator (35 marks)</u>

Looking back on the taxation scenario in stage 1 it wasn't all that "realistic" to be applying the same taxation rate to all taxpayers regardless of their taxable income level, so now we are going to implement a new version of the program from stage 1 which incorporates the following:

- a multi-tier taxation rate system based on the assessable income level (**taxable income minus tax offsets**) of the taxpayer
- an additional levy for taxpayers who do not have private health insurance (Medicare levy)

<u>Changes to the previous program</u>

The multi-tier tax system implements a schedule of increasing tax rates for taxpayers who have received higher levels of assessable income in the previous financial year – this schedule of taxation rates and corresponding tax calculations is arranged as a series of taxation "brackets" as shown below:

Assessable income (taxable income minus tax offsets)	**Tax on this income**
0 - $10,000	Nil
$10,001 - $40,000	15c for each $1 over $10,000
$40,001 - $80,000	$4,500 plus 30c for each $1 over $40,000
$80,001 - $120,000	$16,500 plus 40c for each $1 over $80,000
$120,001 and over	$32,500 plus 45c for each $1 over $120,000

In this updated version of the tax calculation program you should determine which taxation "bracket" the taxpayer's assessable income falls within and proceed to calculate their tax payable amount based on the corresponding taxation formula above.

Once the initial tax payable amount has been calculated the program should proceed to prompt the user as to whether the taxpayer has private health insurance or not. If the user's response is "N" (no) then an additional levy of **1.5%** of the taxpayers taxable income (representing the standard Medicare levy) should be calculated and the result stored for later use.

After the Medicare levy has been calculated (if required) then the program should prompt for the name of the tax agent who will be filing the tax return.

Once tax agent's name has been read in then the program should display a summary of all taxation data to the screen - the Medicare levy figure should only be displayed when taxpayer has had to pay it (ie. the medicare levy itself was calculated to be an amount greater than zero).

Note: You should define constants within your program for all applicable thresholds and tax/levy rates discussed in the specification above.

Sample execution runs showing how the program should function are shown on the following pages.

```
*** Enhanced Income Tax Calculator ***

Enter Name: Bob The Builder

Enter Tax File Number (TFN): 123 456 789

Enter financial year: 2013-2014

Enter assessable income for period 2013-2014: 15000

Enter bank interest accrued: 5000

Enter pre-tax superannuation contribution for period 2013-2014: 3000

Enter claimable deduction(s) for period 2013-2014: 2000

Does the taxpayer currently have private health insurance? (Y/N): N

Enter tax agent name: Mr Fixit

       *** Final Tax Statement ***

Name:                    Bob The Builder
Tax File Number:         123 456 789
Financial Year:          2013-2014

Assessable Income:       $    20000.00
Minus Tax Offsets:       $    -5000.00
Taxable Income:          $    15000.00

Tax Payable:             $      750.00
Medicare levy:           $      225.00

Tax Agent: Mr Fixit
```

Testing tax calculation with medicare levy

Note: Tax Payable figure corrected 13/03/14

Medicare Levy - 1.5% of $15000 = $225

🟦 = user input

🟥 = program variable output

```
*** Enhanced Income Tax Calculator ***

Enter Name: Bob The Builder

Enter Tax File Number (TFN): 123 456 789

Enter financial year: 2013-2014

Enter assessable income for period 2013-2014: 15000

Enter bank interest accrued: 5000

Enter pre-tax superannuation contribution for period 2013-2014: 3000

Enter claimable deduction(s) for period 2013-2014: 2000

Does the taxpayer currently have private health insurance? (Y/N): Y

Enter tax agent name: Mr Fixit

       *** Final Tax Statement ***

Name:                    Bob The Builder
Tax File Number:         123 456 789
Financial Year:          2013-2014

Assessable Income:       $    20000.00
Minus Tax Offsets:       $    -5000.00
Taxable Income:          $    15000.00

Tax Payable:             $      750.00

Tax Agent: Mr Fixit
```

Testing tax calculation without medicare levy

Medicare levy not shown as it does not apply for this taxpayer

🟦 = user input

🟥 = program variable output

```
*** Enhanced Income Tax Calculator ***

Enter Name: Bob The Builder

Enter Tax File Number (TFN): 123 456 789

Enter financial year: 2013-2014

Enter assessable income for period 2013-2014: $75000

Enter bank interest accrued: $10000

Enter pre-tax superannuation contribution for period 2013-2014: $5000

Enter claimable deduction(s) for period 2013-2014: $5000

Does the taxpayer currently have private health insurance? (Y/N): N

Enter tax agent name: Mr Fixit

       *** Final Tax Statement ***

Name:                    Bob The Builder
Tax File Number:         123 456 789
Financial Year:          2013-2014

Assessable Income:       $    85000.00
Minus Tax Offsets:       $   -10000.00
Taxable Income:          $    75000.00

Tax Payable:             $    15000.00
Medicare levy:           $     1125.00

Tax Agent: Mr Fixit
```

Testing tax calculation for a higher bracket with medicare levy

Note that you should test your program more thoroughly than is shown in the sample execution runs above to make sure that it calculates tax correctly for taxable income amounts that fall within each of the tax brackets.

Stage 3 – Final Income Tax Calculator (30 marks)

Now that we've handled taxation rate brackets andthe medicare levy in stage 2 we can move onto making the tax calculator a little more flexible in how it records taxation items to cater for taxpayers who have changed jobs one or more times during the previous financial year.

It is also possible for taxpayers to have more than one bank account (thus accruing interest on multiple accounts), have made contributions to more than one superannuation fund (if they have changed jobs during the year) and it is certainly possible for taxpayers to have multiple tax deductions to submit (as they often just give the tax agent a bunch of receipts and each receipt needs to be recorded one at a time).

To cater for the issues described above you need to take implement a new version of the program developed previously in stage 2 so that the data entry process for taxation items is repetitive instead of sequential, allowing the user to enter figures for each of the different taxation items that need to be recorded as many times as may be required.

To do this the program should first prompt for the basic details (name, tax file number and financial year) as described in stages 1 & 2. After the basic details have been entered by the user the program should then proceed to display a menu which allows the user to select which taxation data item they wish to enter next.

The program should continue to return to the taxation data entry menu until the user chooses to exit the taxation data entry phase of the program's execution, at which time it should then proceed to gather other details for levies/tax agent name and display the final tax statement to the screen as described in stage 2.

A sample execution run demonstrating the structure of the menu itself and how the taxation data entry process should run is shown on the next few pages:

```
*** Final Income Tax Calculator ***
Enter Name: Bob The Builder
Enter Tax File Number (TFN): 123 456 789
Enter financial year: 2013-2014
--------
*** Taxation Data Entry System ***
A - Add assessable income
B - Add interest accrued from bank account
C - Add pre-tax superannuation contribution
D - Add claimable deduction
X - Exit and compile final taxation statement
```

Add PAYG wages

```
Enter your selection: A
Enter assessable income for period 2013-2014: $50000
--------
*** Taxation Data Entry System ***
A - Add assessable income
B - Add interest accrued from bank account
C - Add pre-tax superannuation contribution
D - Add claimable deduction
X - Exit and compile final taxation statement
```

Add bank interest

```
Enter your selection: B
Enter bank interest accrued: $5000
--------
*** Taxation Data Entry System ***
A - Add assessable income
B - Add interest accrued from bank account
C - Add pre-tax superannuation contribution
D - Add claimable deduction
X - Exit and compile final taxation statement
```

Add superannuation contribution

```
Enter your selection: C
Enter pre-tax superannuation contribution for period 2013-2014: $3000
--------
*** Taxation Data Entry System ***
A - Add assessable income
B - Add interest accrued from bank account
C - Add pre-tax superannuation contribution
D - Add claimable deduction
X - Exit and compile final taxation statement
```

Add a deduction

```
Enter your selection: D
Enter claimable deduction(s) for period 2013-2014: $2000
--------
```

🟦 = user input

🟥 = program variable output

Add another deduction

```
*** Taxation Data Entry System ***
A - Add assessable income
B - Add interest accrued from bank account
C - Add pre-pax superannuation contribution
D - Add claimable deduction
X - Exit and compile final taxation statement
Enter your selection: D
Enter claimable deduction(s) for period 2013-2014: $1000
--------
```

Add bank interest from another account

```
*** Taxation Data Entry System ***
A - Add assessable income
B - Add interest accrued from bank account
C - Add pre-pax superannuation contribution
D - Add claimable deduction
X - Exit and compile final taxation statement
Enter your selection: B
Enter bank interest accrued: $1000
--------
```

Add PAYG wages from a second job

```
*** Taxation Data Entry System ***
A - Add assessable income
B - Add interest accrued from bank account
C - Add pre-pax superannuation contribution
D - Add claimable deduction
X - Exit and compile final taxation statement
Enter your selection: A
Enter assessable income for period 2013-2014: $20000
*** Taxation Data Entry System ***
--------
```

Add additional super contributions from that (second) job

```
A - Add assessable income
B - Add interest accrued from bank account
C - Add pre-pax superannuation contribution
D - Add claimable deduction
X - Exit and compile final taxation statement
Enter your selection: C
Enter pre-tax superannuation contribution for period 2013-2014: $2000
--------
```

Data entry finished

```
*** Taxation Data Entry System ***
A - Add assessable income
B - Add interest accrued from bank account
C - Add pre-pax superannuation contribution
D - Add claimable deduction
X - Exit and compile final taxation statement
Enter your selection: X
Taxation data entry complete.
--------
```

🟦 = user input

🟥 = program variable output

```
Does the taxpayer currently have private health insurance? (Y/N): N

Enter tax agent name: Mr Fixit

       *** Final Tax Statement ***

Name:                    Bob The Builder
Tax File Number:         123 456 789
Financial Year:          2013-2014

Assessable Income:       $    76000.00
Minus Tax Offsets:       $    -8000.00
Taxable Income:          $    68000.00


Tax Payable:             $    12900.00
Medicare levy:           $     1020.00

Tax Agent: Mr Fixit
```

Final calculations done and results printed once data entry is complete

🟦 = user input

🟥 = program variable output

Coding Style (5 marks)

Your program should demonstrate appropriate coding style, which includes:

- Levels of 3 or 4 spaces used to indent rather than tabs - you can set up your IDE/editor to automatically replace tabs with levels of 3 or 4 spaces.
- Indentation levels used are consistent throughout program
- A new level of indentation added for each new class/method/ control structure used
- Going back to the previous level of indentation at the end of a class/method/control structure (before the closing brace if one is being used)
- Lines of code not exceeding 80 characters in length - lines which will exceed this limit are split into two or more segments where required
- Expressions are well spaced out and source is spaced out into logically related segments
- Use of appropriate identifiers wherever possible to improve code readability
- Use of comments to describe the purpose of each data class, each method within a data class and any non-trivial segments of code within those methods.

General Implementation Notes/Guidelines

- You may either implement three separate programs (one for each stage), or start off with writing a program for stage 1, then adapt that program to incorporate the additional requirements for stages 2 and 3 in turn.

  The end result we are looking for (ie. what we will actually be marking) will be in the program you implement for stage 3 (which incorporates the requirements set out for stages 1 and 2).

  If you cannot complete all of the requirements for stages 1, 2 and 3 then you should submit a program that implements as many of the requirements as you can manage.

- Your final (submitted) program should be a procedural program consisting of a single class - whilst it is ok to write "helper" methods in the class you are writing, you should refrain from implementing "helper" or "utility" classes or otherwise trying to implement the program across multiple classes in an object-oriented manner.

  You are being assessed on your ability to write a program in a procedural manner in this assignment, not simulate something you would do in the Alice programming environment within a Java program or otherwise demonstrate your object-oriented programming prowess (there will be ample opportunity to demonstrate such prowess in later assignments).

  Your submission **will be penalised** if you go against these guidelines by implementing the program across multiple classes in an object-oriented manner – remember that there is no default "World" class in Java that acts as a starting point for your program.

- You should stick to using the **standard Java API** (**version 1.6 and 1.7 are both ok**) when implementing your program – use of third party API's will mean that the markers will not be able to compile and run your program, **which will result in significant penalties for a "non-functional" program being applied**.

- There is no need to use any complex data structures such as arrays in this task – you are expected to demonstrate the use of simple primitive-type variables and Strings for data storage.

- If you are using the Scanner class for reading input then it is likely that you will run into the "Scanner bug" at some points in the user-input sequence for stages 2 & 3 – the input sequence in these stages was deliberately structured to create a Scanner bug issue at different points and you should deal with this problem in an appropriate manner (this will be discussed during the week 2 tutorial).

Submission tips

If you have developed the program using a text editor and compiling/running manually at a command prompt then you only need to submit the **source code** (.java) file for your program. If you have separate programs for each stage then zip up the corresponding .java files to a plain zip file (ie. not zipx, 7zip, rar, or some other format - just a plain zip file).

If you are using eclipse then you should export your entire eclipse project to a zip archive and submit the resulting zip file - **do this from within eclipse while it is running**, not by fiddling around in the eclipse workspace directly as you may corrupt your entire workspace if you do something wrong. If you are not sure of how to do this then ask – the process of exporting an entire project to an archive file will also be demonstrated during one of the weekly chats.

All students are also advised to check the contents of their zip files by opening them and viewing the files contained within before submitting to make sure they have done it correctly and that the correct (latest) version of the source code file is present to avoid any unpleasant surprises later on.

Submission deadlines / information

This assignment contributes **10%** towards your final result for this course. **This assignment is not a hurdle in/of itself - rather it contributes towards the practical work component for this course (of which you must obtain 50% or better overall in order to pass the course).**

This assignment is due to be submitted via **weblearn** by **11:59pm on Sunday April 6 AEST (end of week 5).**

There will be **a late submission period of 5 days** for this assignment, which will expire at **11:59pm on Friday April 11 AEST.** Late submissions that are received before the end of this late submission period will **attract a late penalty of 10% per day (or part thereof) of the marks awarded on the assignment**, unless an extension has been granted by the school (see below).

Submissions that are received after the late submission period expires at **11:59pm on Friday April 11** will not be assessed unless some prior arrangement has been organised by the **CSIT OUA administrator** (ouacsit@rmit.edu.au) in response to a request for an extension.

Your instructor does not have the authority to negotiate or approve general extensions in this course, so all such requests should be sent to the email address listed above.

Special Note for students who have had EAA provisions organised by the DLU:

If you are a student who has EAA provisions organised by the DLU then you must negotiate any extension you may require with the instructor well in advance of the submission deadline (at least 72 hours in advance of the on-time submission deadline would be ideal).

All questions regarding this specification should be directed to the Assignment 1 Specification forum on blackboard.

Any requests for assistance regarding implementing the requirements set out in this assignment or problems you are having with your program should be directed to the Assignment 1 Help / Discussion forum on blackboard.