Introduction

This assignment gives you practice writing sequences of statements using assignments, method calls, if-statements, and return statements. At the same time, you will get a feel for what people go through in completing their U.S. income tax forms each Spring. You can look at our Java code for GUIs, but you don't have to.

Purpose of this assignment

• Give practice in writing sequences of code using assignment, calls, if-statements, and returns.

• Give practice in reading and understanding already written programs.

• Give more understanding of and practice in the process of programming and testing.

Ground rules

You may work in groups of two. If you work in a group, you must do the work together, sitting at the computer and taking turns managing the keyboard and mouse. It is dishonest and against our academic integrity standards to each write a piece of code, independently, and then to put the finished pieces together.

If you work in a group, form the group on the CMS several days before you submit. One person must invite and the other accept the invitation. Trying to form a group after submitting will waste our time and your time.

You should not show your code or give your code in any form to anyone (except the course staff). You should not look at or be in possession of someone else's code. Of course, you may talk to people about the assignment and discuss solutions in general.

Keep track of how long you spend on this assignment. Just before submitting the assignment, put a comment at the beginning of file F1040.java that says how much time you spent.

Near the end of this document is a section "Hints on writing this assignment". Read it before starting to program.

Idea of this assignment

The form on the last page of this handout was used for people to enter their itemized deductions on Schedule A of their 2008 US Income Tax submission. Each line contains instructions on how to place a number on the righthand side of the line. Some of the instructions can be interpreted as assignment statements, others as function calls, and others as if-statements.

The image on the right is a simplified version of the real Schedule A. The lines on it have essentially the same meaning as those on the real Schedule A. But many lines have been removed, the instructions have been simplified, and some instructions are written partly in terms of Java (e.g. the result of calling ItemizedDeductionWorksheet()).

The image to the right is part of the GUI (Graphical User Interface) we give you, in a Java program. We have already written the Java method that fills in the fields —using a sequence of assignments, function calls, and if-statements. This method was written by translating the instructions on the image into Java.

We ask you to read our class definition for Schedule A, following our hints on how to read and understand it. Then, we ask you to write some methods in the same way for other parts of the income tax calculation.

We use the application TurboTax (not the solution to this assignment, alas!) to fill out our income tax forms each year. It uses well over 50 different forms! Each appears on the monitor at the appropriate time. The user types numbers in a few places, and the application calculates the rest of the fields, sometimes transferring values automatically to other forms, as required by the instructions. In this assignment, you are getting a small taste of what the programmers who developed those applications went through.

Getting started and reading class definition FA.java

From the assignment page of the course webpage, download file `a4.zip`, unzip it, and place the contents in a new directory on your hard drive. (Always start a new project in a new directory.) Load the `.java` files into DrJava and compile.

Now, study class `FA`, for Schedule A, while looking at the image on the previous page. Look at the fields:

Line 3 contains the (name of the) object that describes the Form 1040 with which this Schedule A is connected. Form 1040 is the main form filled out by a taxpayer.

Lines 4..8 contain fields for the values that a taxpayer must enter on Schedule A. Look down at the constructor on line 28 or so, and you will see that these values must be given when a new FA object is created.

Lines 10..22 contain a field for each line number on which a value must be placed on the form (see the image on the previous page). The name of the field is "line" followed by the line number. Our convention is that the line is empty (no number appears on it) if the value of the corresponding field is the minimum **int** value.

Lines 24..35 contains the constructor.

Some values on the lines of Schedule A may be needed in other forms. A call `getLine(i)` of the function defined on lines 37..54 returns the value on line `i`.

Method `fillOutForm`, on lines 56-77, fills in the values on all lines on Schedule A using the instructions given in the image on the previous page —the values get placed in the fields on lines 10-22. We urge you to compare the instructions on the image with the body of the function (lines 59–76); you should understand why each line of the body is the way it is. Only then should you continue reading this document.

Our assignment contains four classes, for four such forms: `F1040`, `FA`, `F6251`, and `W6251`. Each of them has the same information: fields that contain (names of) other forms, perhaps other fields, fields for input values for the form, fields for each line of the form, constructors, a `getLine(i)` function, and a `fillOutForm` method.

Following those items, there may be some more methods and fields that are particular to a form. For example, for Schedule A, there is an "itemized deduction worksheet". Private field `line11OfItemizedDeduction-Worksheet`, function `Line11OfItemizedDeductionWorksheet`, and function `ItemizedDeduction-Worksheet` take care of this itemized deduction worksheet.

The tax forms and corresponding Java files in this assignment

This simplified version of income tax uses four forms. We describe them below, and for each, we mention the two java classes that are responsible for implementing the forms. One class does the computations that have to be performed to compute the values on the form. The other class is the GUI (Graphical User Interface), which displays the form on your monitor —the class extends JFrame. Separating functionality into appropriate classes is an important part of program design. You do *not* have to look at the GUI classes at all, but you can if you want to.

You can see all four forms displayed, with no data in them, by executing `t= new F1040();`.

Form 1040 is the main form that everyone fills out. Class `F1040` is for its computations, and class `GUI1040` provides the GUI.

Schedule A is used by those who want to "itemize their deductions", by stating the state and local taxes they paid, their gifts to charity, their medical expenses, and other things. If one does not itemize (i.e. does not use Schedule A), a standard deduction is used. Class `FA` is for computations, and class `GUIFA` provides the GUI.

Form 6251 is used to calculate an *alternative minimum tax* (AMT), something that almost nobody understands. The 12-page instructions (!) for filling it out says that it applies to taxpayers with certain types of income that receive favorable treatment, and its purpose is to set a limit on the amount by which this favorable treatment can reduce total tax. Put another way, it says, "Hey you are making too much money not to pay so much tax, so pay some more!" This is a good thing, if implemented properly. It used to apply to relatively few people, but inflation since it was created (first in 1969 and in its present form in 1982) has caused more and more people to have to consider this form and be confused.

Class `F6251` is for computations, and class `GUIF6251` provides the GUI.

Worksheet for Form 6251 is used to determine if Form 6251 has to be used. Class `W6251` is for computations, and class `GUIW6251` provides the GUI.

Step 1 of the assignment: Standard Deduction Worksheet

As a first step in writing this assignment, write and test a method to calculate what is called the *standard deduction*. Look at file stdDeductionWorksheet.pdf. It is a worksheet for calculating the standard deduction from two parameters: (1) whether the person is single, married filing separately, or married filing jointly and (2) the amount of taxes placed on line 7. We could have provided a GUI for this worksheet but decided not to because it wouldn't add anything new to the assignment.

In file stdDeductionWorksheet.pdf, we have simplified matters by checking box NO on line 2 and writing 0 on lines 5 and 6.

Your task is to write the body of static method `F1040.standardDeductionWorksheet`. Introduce one local variable for each line of the worksheet (skip lines 2 and 3, because line 2 has box NO checked), and write a statement that assigns the correct value to that variable, literally translating the instruction given in the worksheet. For example, for line 4 you could write

```
int sdwline4= sdwline1;
```

and for line 9, you could write:

```
int sdwline9= Math.min(sdwline7, sdwline8);
```

We ask for a literal translation of the instructions for each line into Java, for two reasons: (1) you get practice in translating into Java and (2) it's the easiest way to get the function correct. We did the same thing in writing method `FA.fillOutForm`.

As specified, the method should return the value that is on line 10.

To help you out, we have written testing function `Tester.testStDeductionWorksheet`.

Step 2 of the assignment. Computing the tax

The main purpose of all the income tax forms is to determine, based on the income, exemptions, taxes already paid, deductions of various kinds, and hundreds of other things, the *taxable income*: the amount on which taxes have to be paid. Pages 68–79 of the instructions contain tables that are used to determine the tax, based on the taxable income. But a taxpayer whose taxable income is above a certain amount has to use a tax-table worksheet to compute their tax, using formulas.

So that we don't have to write 12 pages of tables into our Java program, we will use the tax-table worksheet for *everyone*, regardless of their taxable income. This means that the amount of tax paid will always be 25% or higher, instead of the 10% or even less that people with little income pay.

File taxCompWksht.pdf contains the worksheet. Your task is to write the body of function `F1040.computeTax`, using this worksheet. You don't need local variables. For each possible case (you need at least ten test cases), use a return statement whose expression does the arithmetic required for that case in computing the tax.

To help you out, we have included in `Tester.java` ten test cases that you might use to test method `F1040.computeTax`, but only five have the right expected value. Figure out by hand what the expected value should be for the others. (As of Sunday night Oct 10th, all the test cases in the updated file have the right expected value.)

Step 3 of the assignment. Filling out a worksheet to determine whether Form 6251 has to be filled in

The Worksheet given in file Worksheet6251.jpg is used to determine whether a taxpayer has to complete the alternative minimum tax form. Your job is to complete the body of function `W6251.fillOutForm`.

In completing this function, follow the directions in the Worksheet carefully, assigning to each line as required. You must be painstakingly careful. Here are some hints:

1. To tell whether Schedule A is being used, look for a suitable function in the object given by field `form1040`.

2. You will need some lines from the 1040 form, which is in field `form1040.`

3. To get lines from Schedule A, you have to first get Schedule A, using `form1040.getA()`.

4. The filing status (1, 2, or 3) is on line 1 of the 1040 form —look at it to see what each integer means.

You will have difficulty writing the function correctly unless you work as follows: Write the code to fill in *one line* (or group of lines) at a time, and then test to make sure it is filled in correctly before proceeding. Especially if you are new at this kind of stuff, working in this fashion is the best thing to do.

Finding suitable test cases to test that each line is filled in correctly can be torturous, but it must be done. We help you. Function `Tester.testW6251FillOutForm` creates several new 1040 forms and has the complete testing needed for the first test case. You have to complete the testing of the others. Note that before each assignment that creates a new object there are comments that explain what choices are being examined during the test. These comments help us make sure that each choice given in the worksheet is tested with the choice going one way and the other.

Note the assignment to variable `F1040.formsAreVisible` at the beginning of `testW625FillOutForm`. If you assign `true` to this variable, then the forms will be made visible. But if you do that, comment out all but one of the sets of testing going on in the function, because one set of forms will be created for each one.

Step 4 of the assignment. Completing Form 6251: Alternative Minimum Tax

Look at file Form6251.jpg, which is our form for the alternative minimum tax. It consists of three parts. Part I computes the income on which the alternative minimum tax is based. Part II then computes the tax, based on the result of Part I and part III, an "exemption worksheet". We have simplified the forms! It is almost impossible for the average taxpayer to complete these forms properly without a computer program to do the calculations.

Now look at method `F6251.fillOutForm`, which is completed for you; it calculates part I, calculates Part 2, and returns the value on line 36, as required.

We ask you to complete methods `fillOutPartI` and `fillOutPartII` —we have already completed method `fillOutExemptionWksht()` for you. As you do it, also put suitable testing in class Tester, in a new procedure that you define. You have seen what we did in the testing for step 3; do something like that.

Here are a few hints.

1. To get the value of line 11 of the itemized deduction worksheet, use

    ```
    form1040.getA().Line11OfItemizedDeductionWorksheet();
    ```

2. To get the value found by filling in the exemption worksheet, use the call `fillOutExemptionWksht()`.

3. The instructions for line 36 say that the value on the line also goes on Form 1040 line 45. There's no need to do anything about this; `F6251.fillOutForm`   returns that value and it will be placed on line 45 by a calling method.

You will need some different test cases; here are three you can try.

```
// standard deduction, use exemption worksheet but then it says not to before line 61, line 32 uses
// the multiplication by .28, line 36 < 0.
fa= new F1040("Gries a", 3, 2, 340000, 20000);

// itemized deduction, using exemption worksheet, line 31 > 0, line 36 > 0
fb= new F1040("Gries b", 2, 2, 400000, 3000, 30000, 2500, 3000, 5);

// itemized deduction, exemption say $1.00!,
fc= new F1040("Gries c", 1, 2, 300000, 3000, 60000, 2500, 3000, 5);
```

Hints on writing this assignment

1. On the tax forms, all lines are of type **int**. If **double** numbers are calculated, *round* them to **int**s before putting them on a line.

2. Make use of `Math.max` and `Math.min`. For example, an expression like "line 24 - line 26; if < 0, enter 0" can be evaluated using `Math.max(line24 - line26, 0)`.

3. In the tester, you can suppress showing of forms on the monitor by executing

    ```
    F1040.formsAreVisible= false;
    ```

    Static variable `formsAreVisible` is public so that it can be easily changed.

4. Follow directions about how to develop. Write one line at a time and test it, if necessary; don't go on until you   are sure it is right. We see too many people writing lots of code, figuring they will test it later. These leads to huge amounts of wasted time, as people try to track down complex bugs that would have been easy to fix if the initial bugs had been tested for (and thus caught) earlier.

5. In figuring out the results of test cases, figure them out by hand. We did that, and in a few cases we actually found some typos caused by putting digits in the wrong order in numbers, etc.

6. Write test cases for a method first (in class Tester); then write the method body.

Completing the assignment

Be sure your java files are indented properly.

Submit files `F1040.java`, `W6251.java`, and `F6251.java`, which are the only ones you change, on the CMS by the end of Saturday, 16 October.

SCHEDULES A&B
(Form 1040)

Department of the Treasury
Internal Revenue Service   (99)

Schedule A—Itemized Deductions

(Schedule B is on back)

▶ Attach to Form 1040.   ▶ See Instructions for Schedules A&B (Form 1040).

OMB No. 1545-0074

2008

Attachment
Sequence No. **07**

Name(s) shown on Form 1040

Your social security number

Medical and Dental Expenses	**Caution.** Do not include expenses reimbursed or paid by others.		
	1 Medical and dental expenses (see page A-1).	**1**	
	2 Enter amount from Form 1040, line 38 **2**		
	3 Multiply line 2 by 7.5% (.075)	**3**	
	4 Subtract line 3 from line 1. If line 3 is more than line 1, enter -0-.	**4**	
Taxes You Paid (See page A-2.)	**5** State and local **(check only one box):**		
	a ☐ Income taxes, **or**	**5**	
	b ☐ General sales taxes		
	6 Real estate taxes (see page A-5)	**6**	
	7 Personal property taxes .	**7**	
	8 Other taxes. List type and amount ▶	**8**	
	9 Add lines 5 through 8	**9**	
Interest You Paid (See page A-5.)	**10** Home mortgage interest and points reported to you on Form 1098	**10**	
	11 Home mortgage interest not reported to you on Form 1098. If paid to the person from whom you bought the home, see page A-6 and show that person's name, identifying no., and address ▶		
Note. Personal interest is not deductible.		**11**	
	12 Points not reported to you on Form 1098. See page A-6 for special rules.	**12**	
	13 Qualified mortgage insurance premiums (see page A-6)	**13**	
	14 Investment interest. Attach Form 4952 if required. (See page A-6.)	**14**	
	15 Add lines 10 through 14	**15**	
Gifts to Charity If you made a gift and got a benefit for it, see page A-7.	**16** Gifts by cash or check. If you made any gift of $250 or more, see page A-7	**16**	
	17 Other than by cash or check. If any gift of $250 or more, see page A-8. You **must** attach Form 8283 if over $500	**17**	
	18 Carryover from prior year	**18**	
	19 Add lines 16 through 18	**19**	
Casualty and Theft Losses	**20** Casualty or theft loss(es). Attach Form 4684. (See page A-8.)	**20**	
Job Expenses and Certain Miscellaneous Deductions (See page A-9.)	**21** Unreimbursed employee expenses—job travel, union dues, job education, etc. Attach Form 2106 or 2106-EZ if required. (See page A-9.) ▶	**21**	
	22 Tax preparation fees	**22**	
	23 Other expenses—investment, safe deposit box, etc. List type and amount ▶	**23**	
	24 Add lines 21 through 23	**24**	
	25 Enter amount from Form 1040, line 38 **25**		
	26 Multiply line 25 by 2% (.02)	**26**	
	27 Subtract line 26 from line 24. If line 26 is more than line 24, enter -0-	**27**	
Other Miscellaneous Deductions	**28** Other—from list on page A-10. List type and amount ▶	**28**	
Total Itemized Deductions	**29** Is Form 1040, line 38, over $159,950 (over $79,975 if married filing separately)?		
	☐ **No.** Your deduction is not limited. Add the amounts in the far right column for lines 4 through 28. Also, enter this amount on Form 1040, line 40. } ▶	**29**	
	☐ **Yes.** Your deduction may be limited. See page A-10 for the amount to enter.		
	30 If you elect to itemize deductions even though they are less than your standard deduction, check here ▶ ☐		

For Paperwork Reduction Act Notice, see Form 1040 instructions.   Cat. No. 11330X   **Schedule A (Form 1040) 2008**