# “To Itemize or Not To Itemize- that is the question”

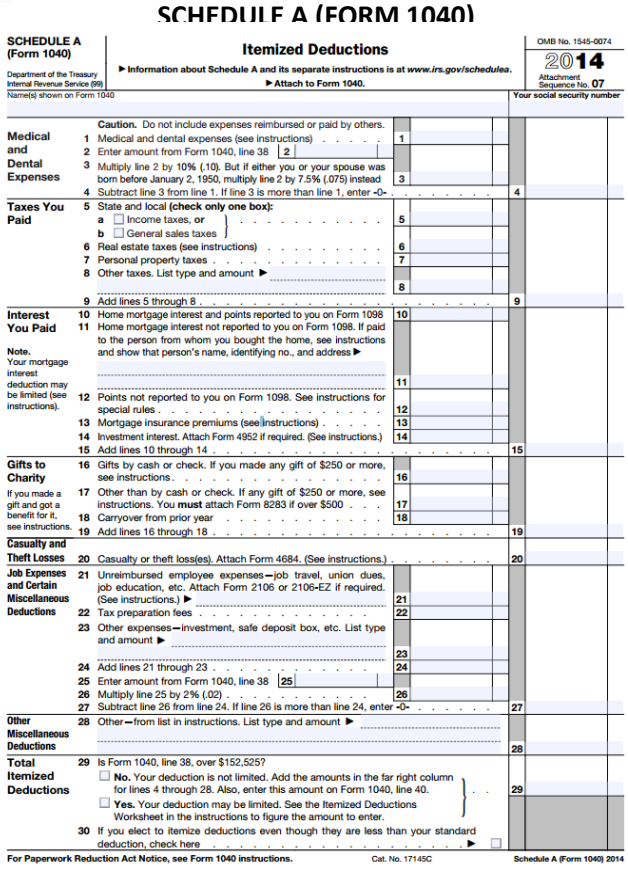
**By**
**Calvin Smith**

## Executive summary

Every year millions of individuals and households are required to file an income tax return. However, much of the general public is unaware of how to optimize their tax return. Tax filing services, automated or provided by an accountant, can be expensive and time consuming. From my experience as a VITA (Volunteer Income Tax Assistance) volunteer for three years, I’ve found that one component of the tax return that can be especially time consuming is determining if it is more beneficial for the taxpayer to itemize deductions or to take the standard deduction. Navigating Schedule A (Form 1040) can be a frustrating experience, even for an accountant. Finding relevant tax guidance that corresponds with each line item and calculating phase outs and limitations can be extremely time intensive. As a result, individuals who may be better off itemizing their deductions often take the standard deduction to avoid the frustrating and confusing forms.

My VBA project provides an easy to use calculator that not only determines which method (stardard or itemized) will be most beneficial for the tax payer given their specific situation, but also provides instructions and guidance to help the tax payer enter the correct inputs into the calculator. The VBA project essentially walks the user through all potential deductions from Schedule A and applies the relevant calculations to determine the amount the user can itemize. Then, the program compares the total elgible itemized deductions against the standard deduction and recommends most beneficial method (standard or itemized) of tax filing.

A second potential application of this calculator exists. When tax payers are considering making a major tax planning decision that will impact how they itemize, like whether they should rent or buy a home that will have tax deductible interest, users can run a variety of scenarios to see how that tax planning strategy will impact their itemized deduction. The IRS does not require tax payers to pay more taxes than they are legally obligated to pay. The purpose of this project is not to promote tax evasion, but to help taxpayers structure their financial decisions in a more tax efficent manner and more easily utilize available deductions the IRS has made available.



SCHEDULE A (Form 1040)

SCHEDULE A (Form 1040) — Itemized Deductions — OMB No. 1545-0074 — 2014

Department of the Treasury, Internal Revenue Service (99)

▶ Information about Schedule A and its separate instructions is at www.irs.gov/schedulea.

▶ Attach to Form 1040.

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 instructions) | 1 | |
| | 2 | Enter amount from Form 1040, line 38 [2] | | |
| | 3 | Multiply line 2 by 10% (.10). But if either you or your spouse was born before January 2, 1950, multiply line 2 by 7.5% (.075) instead | 3 | |
| | 4 | Subtract line 3 from line 1. If line 3 is more than line 1, enter -0- | | 4 |
| Taxes You Paid | 5 | State and local (check only one box): a ☐ Income taxes, or b ☐ General sales taxes | 5 | |
| | 6 | Real estate taxes (see instructions) | 6 | |
| | 7 | Personal property taxes | 7 | |
| | 8 | Other taxes. List type and amount ▶ | 8 | |
| | 9 | Add lines 5 through 8 | | 9 |
| Interest You Paid | 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 instructions and show that person's name, identifying no., and address ▶ | 11 | |
| Note. Your mortgage interest deduction may be limited (see instructions). | 12 | Points not reported to you on Form 1098. See instructions for special rules | 12 | |
| | 13 | Mortgage insurance premiums (see instructions) | 13 | |
| | 14 | Investment interest. Attach Form 4952 if required. (See instructions.) | 14 | |
| | 15 | Add lines 10 through 14 | | 15 |
| Gifts to Charity | 16 | Gifts by cash or check. If you made any gift of $250 or more, see instructions | 16 | |
| If you made a gift and got a benefit for it, see instructions. | 17 | Other than by cash or check. If any gift of $250 or more, see instructions. 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 instructions.) | | 20 |
| Job Expenses and Certain Miscellaneous Deductions | 21 | Unreimbursed employee expenses—job travel, union dues, job education, etc. Attach Form 2106 or 2106-EZ if required. (See instructions.) ▶ | 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 in instructions. List type and amount ▶ | | 28 |
| Total Itemized Deductions | 29 | Is Form 1040, line 38, over $152,525? ☐ 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. ☐ Yes. Your deduction may be limited. See the Itemized Deductions Worksheet in the instructions to figure the amount to enter. | | 29 |
| | 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. 17145C Schedule A (Form 1040) 2014

**Implementation documentation**

**Design and Layout of User Interface and Supporting Inputs:** The first component of the implementation required that I create an interface with which the user would interact with the program. I wanted an interface that could be used by inexperienced tax payers, but be recognizable by sophisticated users. For these reasons, the user interface follows a pattern that reflects the structure and line items on a Schedule A (Form 1040).

I decided against creating a form based application so that more experienced users to easily insert formulas and calculated values into the calculator as well as perform side calculations on the same page as the program. Ease of use or functionality are not impaired by my non-form approach and no excel formulas are used in the cells that form my calculator. Using VBA to automate this process rather than formulas in cells has the advantage of making the calculator more robust for inexperienced users. The use of VBA allows simple operation of the calculator and generates a clear answer to the question: "Should I itemize?"

**Itemize or don't Itemize?** Tax Year: 2014

| | | | |
|---|---|---|---|
| **Medical and dental Expenses** | | | |
| 1 | Medical and dental expenses | $5,001.00 | HELP |
| 2 | Adjusted gross income | $50,000.00 | HELP |
| 3 | Calculated Value | $3,750.00 | |
| 4 | Total deductible medical and dental exp | $1,251.00 | |
| **Taxes you Paid** | | | |
| 5 | State and local | | |
| | a. income taxes, or | $ 1.00 | HELP |
| | b. General sales taxes | $ 1.00 | |
| 6 | Real estate taxes | $ 1.00 | HELP |
| 7 | Personal property taxes | $ 1.00 | HELP |
| 8 | Other taxes | $ 1.00 | HELP |
| 9 | Total taxes paid | $5.00 | |
| **Interest you paid** | | | |
| 10 | Home mortgage interest and points repo | $ 1.00 | HELP |
| 11 | Home mortgage interest not report | $ 1.00 | |
| 12 | Points not reported on form 1098 | $ 1.00 | HELP |
| 13 | Mortgage insurance premiums | $ 1.00 | HELP |
| 14 | Investment interest | $ 1.00 | HELP |
| 15 | Total interest paid | $5.00 | |
| **Gifts to charity** | | | |
| 16 | Gifts by cash or check | $ 1.00 | HELP |
| 17 | Other than by cash or check | $ 1.00 | HELP |
| 18 | Carry over from prior year | $ 1.00 | HELP |
| 19 | Total gifts to charity | $3.00 | |
| **Casualty and theft losses** | | | |
| 20 | Casualty or theft losses | 50 | HELP |
| **Job expenses and certain miscellaneous deductions** | | | |
| 21 | Unreimbursed employee expenses | $ 1.00 | HELP |
| 22 | Tax preparation fees | $ 1.00 | HELP |
| 23 | other expenses | $ 1.00 | HELP |
| 24 | Sub total calculation | $3.00 | |
| 25 | Adjusted gross income | $50,000.00 | |
| 26 | Calculated value | $1,000.00 | |
| 27 | Total deductible miscellaneous deductic | $0.00 | |
| **Other miscellaneous deductions** | | | |
| 28 | Other-from list in instructions | 1 | HELP |
| **Total itemized deductions** | | | |
| 29 | Total itemized if not limited | $1,315.00 | |
| | Total itemized if limited | | |
| | Your Standard deduction | $12,400.00 | |
| | Your itemized deduction | $1,315.00 | |
| | Should I itemize? | No | |

Because I wanted the calculator to be able to be easily modified between tax years with new standard deduction rates and new information, I created a supporting information tab. This tab allows someone without VBA experience, but some tax experience, to quickly update key inputs like standard deduction rates and line description/help information. This prevents this calculator from becoming obsolete once tax season is over.

After volunteering with BYU's Volunteer Income Tax Assistance lab for three years, I've found that one of the most significant challenges for tax preparers filling out IRS forms is understanding what goes in each box of the form. To address this issue, I designed the form with help buttons that immediately bring the applicable tax guidance for the line they need help with into the calculator. This tax guidance was drawn directly from the IRS website which provides instructions for filling out Schedule A 1040. This allows users to view tax help as they need it and prevents users from having to look up the guidance themselves and having to tab between the tax guidance and the calculator.

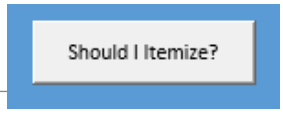
The last component of the user interface worksheet is the personal information box. As I looked into Schedule A, I found that a taxpayer's age can impact how deductions are calculated. To make sure users enter valid inputs, I implemented sheet level validity checks to ensure that information would transfer to the VBA variables correctly. Because a tax payer may not readily understand filing statuses available, and because I did not use a vba form, I utilized a drop down menu to ensure the taxpayer or user knows the available filing statuses and to ensure the filing status entered is valid before the VBA code is executed. I opted to use a drop down menu rather than an ActiveX combo box because the user may not have ActiveX controls or software enabled on their computer. The use of a traditional excel drop down box ensures a more seemless user experience for more users.

**Personal information**

| | Year Born | |
|---|---|---|
| Your Year of birth | | 1949 |
| Spouse Year of birth | | 1955 |
| Adjusted Gross Income | $ | 50,000.00 |
| Filing Status | | Married filing jointly |

**VBA Implementation:** The core value that my VBA implementation pivoted around was understability and future proofing. I know that tax laws change over time and with those changes my calculator will need to change in order to stay relevant to the tax year in which it is used. I wanted other VBA users to be able to understand what my program was doing, in what order, and why it was doing it without the extensive use of comments or a sophistocated user guide. For example: The numbers of the variables in the code correlate to the number line both in the calculator and on a Schedule A for understandability purposes. For example: "one" in sub medical is line 1 in the calculator user interface and line 1 in the Schedule A. This patter carries throughout the calculator.

```
Sub update()

Dim taxyear As Integer
Dim x As Integer
agi = Worksheets("worksheet").Range("J5").Value
yob1 = Worksheets("worksheet").Range("J3").Value
yob2 = Worksheets("worksheet").Range("J4").Value
taxyear = Worksheets("Supporting Information").Range("G5").Value
Worksheets("Worksheet").Range("F1").Value = taxyear

For x = 0 To 4
Range("G48").Offset(x, 0).Style = "currency"
Range("G48").Offset(x, 0) = ""
Next

Call filingstatus
Call medical
Call taxes
Call interest
Call gift
Call job
Call total
Call limittest
Call yesno

If Range("G52").Value = "No" Then
    MsgBox ("No")
Else
    MsgBox ("Yes")
End If

End Sub
```




While I designed the standard deduction and help information to be easily updated year to year, the rest of the mechanics of the Schedule A are built into the VBA of the calculator. While the mechanics of the VBA will not be as easy to adjust as the standard deduction information and help information, the structure of the main code in module 1 should be readily apparent to a novice VBA user.

The update sub runs the entire calculator and is organized according to the structure of the Schedule A 1040. The update sub is triggered by the user pressing the "Should I Itemize?" button. The button was labeled this way to prevent confusion regarding how the calculator should be initiated. Variables are defined at the start of the sub and formatting is applied. The sub then executes subs that correspond to the calculations required by each section of the Schedule A. The subs are given names that relate to the sections on the Schedule A so that another user with some understanding of Schedule A and VBA experience can easily identify and make required adjustments from year to year. After the required calculations are performed, the sub provides the answer to the question "Should I Itemize?" in a message box, answering "Yes" or "No."

Descriptions of what each of the components/subs do in the calculator are listed below:

***Please note: Some Totals will not carry through when you run the calculator depending on your input values. THIS IS NOT A FLAW OR ERROR in the VBA code. Some tax benefits are not realized in the itemizing process until certain calculated thresholds are reached. In some cases Total lines for a category will show ZERO even though expenses are entered in the related category.**

Sub filingstatus: Sub filingstatus looks at the valued entered for filing status and retrieves the corresponding standard deduction value from the "Supporting Information" tab and drops that value into the Total itemized deductions section of the calculator so the viewer can compare their calculated itemized deduction against their standard deduction. This item is retrieved from a table on the "Supporting Information" tab because the user may not readily know what their standard deduction may be given their filing status.

```
Sub filingstatus()
Dim standarddeduction As Integer
Dim status As String

status = Range("J6").Value

Select Case status
Case "Married filing seperately"
    Worksheets("Worksheet").Range("G50").Value = Worksheets("Supporting information").Range("D9").Value
Case "Single"
    Worksheets("Worksheet").Range("G50").Value = Worksheets("Supporting information").Range("D6").Value
Case "Head of Household"
    Worksheets("Worksheet").Range("G50").Value = Worksheets("Supporting information").Range("D7").Value
Case "Married filing jointly"
    Worksheets("Worksheet").Range("G50").Value = Worksheets("Supporting information").Range("D8").Value

End Select
End Sub
```

Sub medical: Sub medical looks at the year of birth information captured in the update sub to determine if standard deduction rates apply to medical expenses. Different tax rules apply for individuals who either are born before 1950 or have a spouse they are filing with that was born before 1950. This code looks at the years during which the taxpayer and spouse (if applicable) were born to determine which taxpayer is oldest and if the oldest person was born before 1950. Without getting too tax technical in this write-up, I'll just say that the appropriate rates are applied according to age. The amount you can itemize from medical expenses is limited, so a calculation is applied to lines one and three to provide line four and line four is limited to positive numbers. **If the calculated threshold is not exceeded, no tax benefit is realized.** The numbers of the variables correlate to the number line both in the calculator and on a Schedule A for understandability purposes. For example: "one" in

```
Sub medical()
Dim one As Currency
Dim two As Currency
Dim three As Currency
Dim four As Currency
Dim oldest As Integer

one = Range("G4").Value
two = agi

If yob1 < yob2 Then
oldest = yob1
Else
oldest = yob2
End If

If oldest < 1950 Then
three = two * 0.075
Else
three = two * 0.1
End If

four = one - three

If four < 0 Then
four = 0
End If

Worksheets("Worksheet").Range("G4").Value = one
Worksheets("Worksheet").Range("G5").Value = two
Worksheets("Worksheet").Range("G6").Value = three
Worksheets("Worksheet").Range("G7").Value = four


End Sub
```

sub medical is line 1 in the calculator user interface and line 1 in the Schedule A. This pattern carries throughout the calculator. The last lines of sub medical return the calculated values back to the calculator so the user can use them.

Subs for taxes, interest, and gift: Sub taxes, sub interest, and sub gift are all simple routines because the expenses incurred by these amounts paid is not a limited itemized deduction, so the VBA in these subs simply adds the values preceding the total line to calculate the line total value. The addition takes place in a loop that collects the number values from the applicable cells an adds them to a running total held in the nine variable.

```
Sub taxes()
Dim x As Integer
Dim row As Integer
Dim nine As Currency

nine = 0

For x = 0 To 4
Range("G11").Offset(x, 0).Style = "currency"
nine = nine + Range("G11").Offset(x, 0)
Next

Range("G16") = nine

End Sub
```

Sub job: Sub job is related to the "Job expenses and certain miscellaneous deductions section of the calculator. These expenses are limited itemized deductions, up to 2 percent of adjusted gross income. This code takes the module level variable, AGI and applies the appropriate rate to determine the extent to which job expenses and miscellaneous deductions are limited or deductible. **If the sum of the deductions in this category does not exceed the threshold, no tax benefit is realized.** The existance of these kinds of expense/deductibility limitations is one of the reasons why itemizing can be so difficult to determine. From my experience in BYU's Volunteer Income Tax Assistance lab, I've found that most tax preparers do not understand how limited deductibility works on the Schedule A. This calculator takes care of this calculation for the tax preparer, and prevents the preparer from deducting nondeductible portions of job or medical expenses. Without these calculations in place, many tax preparers would likely deduct more than is permissible by tax law and incorrectly determine that it would be more beneficial for them to itemize when they shouldn't.

```
Sub job()
Dim x As Integer
Dim twentyfour As Currency
Dim twentysix As Currency
Dim twentyseven As Currency
twentyfour = 0

For x = 0 To 2
Range("G36").Offset(x, 0).Style = "currency"
twentyfour = twentyfour + Range("G36").Offset(x, 0)
Next

Range("G39") = twentyfour

Range("G40") = agi
twentysix = agi * 0.02
twentyseven = twentyfour - twentysix

Range("G41") = twentysix

If twentyseven < 0 Then
twentyseven = 0
End If

Range("G42") = twentyseven



End Sub
```

Sub total: Sub total simply adds all the sub totals to create an unadjusted itemized total. This total can be further limited by additional adjustments depending on the taxpayers AGI in the limittest sub, discussed below.

Sub limittest: The limittest sub is one of the most complicated calculation sections of the VBA of this project. I wasn't able to fit a screen shot of it on this page, so look at my VBA if you want to see it. As far as VBA coding goes, it is not complicated, but the combination of calculations and tests it performs were complicated to implement together. In short, this code determines if the amount the taxpayer wants to itemize is limited. To create this code I based my code on a PDF a worksheet the IRS provides tax preparers to determine if the itemized amount is limited. I followed the calculations on the IRS worksheet and coded them into the limit test. The limit test looks at a variety of inputs provided by the user earlier in the calculator to determine if the itemized deduction is limited, and to what extent it is limited if it is limited.

Sub yesno: the yesno is a simple sub that determines for the tax preparer if they should itemize or not by comparing their itemized deduction against their standard deduction. The only complication in this sub arrises from the fact that the user's itemized deduction may be limited, and that limitation must be taken into account in determining if the user should itemize or not. The nested Ifs are necessary in this process because of the variability between scenarios in the instance that the deduction is limited. The calculator breaks out the itemized deduction in terms of limited and unlimited so the user can see if a higher AGI is causing their itemized deduction to be limited. If I did not break out these two items, the nested if statements would be unnessary.

```
Sub yesno()

If Range("G49").Value = "" Then
    Range("G51").Value = Range("G48").Value

    If Range("G48").Value > Range("G50").Value Then
    Range("G52").Value = "Yes"
    Else
    Range("G52").Value = "No"
    End If
Else
    Range("G51").Value = Range("G49").Value
    If Range("G51").Value > Range("G50").Value Then
    Range("G52").Value = "Yes"
    Else
    Range("G52").Value = "No"
    End If
End If
End Sub
```

## Learning and Conceptual Difficulties

**Organization and coding process:** Beyond the projects I created for class, I have not had any experience writing VBA for large projects. The most significant thing I learned while doing this project is how to take an idea for a project or finished product and break it into parts for the building process. My first instinct was to build the entire project into one sub procedure that would run. However, the more I looked at the Schedule A, the more I realized I could break the project into several different sub procedures that could be independently tested implemented. The Schedule A provides several natural breaks for sub procedures with each section requiring its own calculations depending on the limits, phase outs, and requirements in place. After walking through the manual calculations required for a Schedule A, I settled on using my current code structure. From this project and organization process, I learned that a deep understanding of the underlying function or process you are trying to automate is a significant contributor to VBA project's success.

After I determined the best way to organize my project code, I started coding. I learned how to effectively organize and step through VBA processes by tracing the manual process. The manual process as directed on the IRS Schedule A and supporting worksheets, though tedious, was fairly efficient. While I coded this project, I learned that I could recreate the manual process with code

fairly simply with a few significant exceptions. After a few failed attempts to optimize the code beyond the manual process the IRS had set forth, I realized it was unnecessary to reinvent the Schedule A calculation process with code. While this may not be true in every VBA application and automation of manual processes, coding the manual processes into VBA turned out to be the most efficient and accurate way for me to complete the project.

While coding in VBA I often found that the simplest methods created the best results. For example: I experimented with a variety of ways to bring supporting help information to each line. I tried bringing the information in using a text box, pasting the information with a text box, linking the information to the relevant line values, and copying and pasting the information with code. I found that the simplest method, recording myself deleting previous content, copying the relevant information, and pasting it in the correct location provided me with the best results. Now my 18 help buttons work reliably and consistently paste relevant help information. Although I was disappointed I wasn't able to get other data retrieval methods to work as well, I'm glad I found a reliable solution. Copying and pasting the help data from the Supporting Information tab allows users to edit the help and support data with additional help information and updated tax advice without needing to have a knowledge of VBA. Lowering that bar for usability makes this tool more accessible to more people, which will hopefully contribute to greater implementation success.

**Coding difficulties:** The most difficult part of the project for me to code was the sub limittest. I almost didn't build this section of the project because it only applies to the top 5% of tax payers, payers who would likely be using a tax professional rather than this tool, but I wanted to make my deduction calculator robust. This section of code was difficult because the IRS itemized deduction limitation test requires quite a few inputs that can vary depending on the taxpayer's unique situation. I had to create code that could take each unique situation into account. I ended up using six "if" statements and one case select statement to deal with all the complexities that accompany the phase outs and different filing statuses involved in the itemization limitation calculation.

Once I finished the code, I checked for accuracy by manually running the same calculations using the IRS's form for the test. I had to adjust my code seven times to make the code provide the correct values and provide a limited itemized deduction from an unlimited itemized deduction input. Although few people may ever require this function, I feel proud of being able to make such a complicated piece of code work properly and knowing that my calculator would be able to provide an accurate itemizing estimate, even in those unlikely conditions.

**Making the calculator with multiple applications that is adjustable:** One of my most important objectives with this project was creating a project that could be used by most people and a project that could have multiple uses. I learned that by making a few adjustments to my VBA project I could make my project have multiple applications. My project not only helps people determine if they should itemize or not, but has the capability to help people plan tax efficient financial transactions. For example, someone contemplating buying a house could see the effect the purchase would have on their itemized deduction. By entering the estimated deductible interest payments from their proposed home mortgage, they can see what tax benefit they might receive from owning, rather than renting a home. Another tax planning application may be a family trying to determine the effect a charitable contribution will have on their taxes owed at the end of the year, or determining to what extent their medical expenses will be deductible if they itemize.

I also learned how to make sure my VBA project would be useful to a third group of people. While this project will help users plan tax efficient financial actions and determine if they should itemize, this tool will also help individuals to fill out an actual Schedule A should they decide to itemize. Once the user has used the help functions to fill out the calculator and used the VBA subs to calculate phase outs, limited deductibility, and the total itemized deduction; the user has all the information they need in an easily transferable form to complete a Schedule A. One of the benefits to structuring the user interface like a Schedule A is that novice and advanced users alike can readily transfer the calculators' input AND output values to an IRS Schedule A. This makes preparing an itemized deduction much easier. I anticipate some users who already know they should itemize may use this form to help them fill out the Schedule A for this very reason.

The last thing I learned conceptually was how to make a VBA project more adjustable and future proof. While hard coding values into VBA may be sleeker in executing the code and viewing the workbook, dynamic values allow for a calculator that will remain relevant longer. For example: I could have hard coded the standard deductions and help sections in VBA for 2014, but instead I created an information tab that allows users without any VBA experience to adjust these values between tax years. Adding this functionality helps future proof my project.

**Assistance: None other than PDF IRS schedules, worksheets, forms and instructions**

I received no assistance with this project from professor, student, or otherwise except from the IRS website. By providing PDF forms, schedules, instructions, and worksheets relevant to itemizing deductions, the IRS website was immensely helpful. From the IRS forms I developed my user interface, from the IRS instructions I built my understanding of the underlying processes and problems, and from the IRS worksheets I was able to step through calculations which would have been incomprehensible without additional IRS clarification.