Test Question / if-else

TODO

A Few Questions

While loop.

Question: What is a good example of using a while loop where a for loop would make it much more difficult.

```
 1:
 2: # Example of why to use a "while" - reverse a string
 3:
 4: vIn = "abcd"
 5: vOut = ""
 6: i = len(vIn)
 7: while ( i > 0 ):
 8:     i = i - 1
 9:     vOut = vOut + vIn[i]
10:
11: print ( "vOut = ->{}<-".format(vOut ) )
12:
```

or with a different way of offsetting to 0

```
 1:
 2:
 3: # Example of why to use a "while" - reverse a string
 4:
 5: vIn = "abcd"
 6: vOut = ""
 7: i = len(vIn)-1
 8: while ( i >= 0 ):
 9:     vOut = vOut + vIn[i]
10:     i = i - 1
11:
12: print ( "vOut = ->{}<-".format(vOut ) )
13:
```

Formatting.

Question: Is the `{}` a dictionary in the format statement.

```
>>> a = 1.2345
>>> print ( "X Decimal Places {}".format(a) )
>>> print ( "2 Decimal Places {:.2f}".format(a) )
>>> print ( "In Order {} second {} third {}".format( "1st", 2, "last" ) )
```

A better if-else example

Personal Income Tax Calculator. This is not all of taxes. This is just in the case where you have a job and you get a paycheck. Let's say you have $88,000.00 a year in pay. What do the "tax" calculations mean.

First there is a standard deduction. For 2021 this is:

Amount	Description
$12,550	single taxpayers.
$12,550	married taxpayers filing separately.
$18,800	heads of households.
$25,100	married taxpayers filing jointly.

This looks like a table we can turn into an "if"/"else" in python.

```
 1:
 2: print ( "1 for single taxpayers. " )
 3: print ( "2 for married taxpayers filing separately.  " )
 4: print ( "3 for heads of households.  " )
 5: print ( "4 for married taxpayers filing jointly. " )
 6:
 7: maritul_status = input()
 8:
 9: standard_decuction = 12550
10: if maritul_status == "1" :
11:     standard_decuction = 12550
12: elif maritul_status == "2" :
13:     standard_decuction = 12550
14: elif maritul_status == "3" :
15:     standard_decuction = 18800
16: elif maritul_status == "4" :
17:     standard_decuction = 25100
18: else:
19:     print ( "invalid input, should be 1, 2, 3, or 4" )
20:
21: print ( "Standard Deduction = {}".format( standard_decuction ) )
```

```
22:
23:
```

The standard deduction is take off of your income before you calculate your taxes. So the $88,000.00 minus $25,100 is: $62900.

This is the amount we use in the 2nd tax calculation.

If you search for “tax tables 2021” you get:

Tax Rate	Taxable Income Bracket	Tax Owed
10%	$0 to $14,200	10% of taxable income
12%	$14,201 to $54,200	$1,420 plus 12% of the amount over $14,200
22%	$54,201 to $86,350	$6,220 plus 22% of the amount over $54,200
24%	$86,351 to $164,900	$13,293 plus 24% of the amount over $86,350

What this table means is that you pay 10% on the first $14,200. Then take that off then pay 12% on the next chunk of money.

Let’s implement that.

```
 1:
 2: print ( "What is your per year income" )
 3: income_str = input()
 4: income = float(income_str)
 5:
 6: print ( "1 for single taxpayers. " )
 7: print ( "2 for married taxpayers filing separately.  " )
 8: print ( "3 for heads of households.  " )
 9: print ( "4 for married taxpayers filing jointly. " )
10:
11: maritul_status = input()
12:
13:
14: #| Amount  | Description                          |
15: #|---------|--------------------------------------|
16: #| $12,550 | single taxpayers.                    |
17: #| $12,550 | married taxpayers filing separately. |
18: #| $18,800 | heads of households.                 |
19: #| $25,100 | married taxpayers filing jointly.    |
20:
21: tax = 0
22: standard_decuction = 12550
23:
24: if maritul_status == "1" or maritul_status == "2" : # Single, Married File Separat
```

```
25:     standard_decuction = 12550
26:     income = income - standard_decuction
27:
28:     tax = (10/100) * income
29:     if income >= 9951:
30:         tax = tax + (2/100) * ( income - 14200 )
31:     if income >= 40526:
32:         tax = tax + (10/100) * ( income - 54200 )
33:     if income >= 86376:
34:         tax = tax + (2/100) * ( income - 86350 )
35:     if income >= 164926:
36:         tax = tax + (10/100) * ( income - 164925 )
37:     if income >= 209426:
38:         tax = tax + (3/100) * ( income - 209425 )
39:     if income >= 523601:
40:         tax = tax + (2/100) * ( income - 523600 )
41:
42: elif maritul_status == "3" :        # Head of Household
43:     standard_decuction = 18800
44:
45:     income = income - standard_decuction
46:
47:     tax = (10/100) * income
48:     if income >= 14201:
49:         tax = tax + (2/100) * (income - 14200)
50:     if income >= 54201:
51:         tax = tax + (10/100) * (income - 54200)
52:     if income >= 86351:
53:         tax = tax + (2/100) * (income - 86350)
54:     if income >= 164901:
55:         tax = tax + (10/100) * (income - 164900)
56:     if income >= 209401:
57:         tax = tax + (3/100) * (income - 209400)
58:     if income >= 523601:
59:         tax = tax + (2/100) * (income - 523501)
60:
61: elif maritul_status == "4" :
62:     standard_decuction = 25100
63:
64:     income = income - standard_decuction
65:
66:     tax = (10/100) * income
67:     if income >= 14201:
68:         tax = tax + (2/100) * (income - 14200)
69:     if income >= 54201:
70:         tax = tax + (10/100) * (income - 54200)
71:     if income >= 86351:
72:         tax = tax + (2/100) * (income - 86350)
73:     if income >= 164901:
74:         tax = tax + (10/100) * (income - 164900)
75:     if income >= 209401:
```

```
 76:         tax = tax + (3/100) * (income - 209400)
 77:     if income >= 523601:
 78:         tax = tax + (2/100) * (income - 523501)
 79:
 80: else:
 81:     print ( "invalid input, should be 1, 2, 3, or 4" )
 82:
 83: print ( "Standard Deduction = {}".format( standard_decuction ) )
 84:
 85:
 86:
 87:
 88: print ( "total tax for the year = {:.2f}".format(tax) )
 89:
 90: print ( "What is monthly witholding" )
 91: withold_str = input()
 92: withold = float(withold_str)
 93:
 94: owe = tax - ( 12 * withold )
 95: if owe < 0 :
 96:     print ( "You get a tax refund of {:.2f}".format(-owe) )
 97: elif owe == 0:
 98:     print ( "You don't owe any and you don't get a refund" )
 99: elif owe > 0:
100:     print ( "Send the IRS: {:.2f}".format(owe) )
101:
102:
```

Import

Most of the time when you build a program you have multiple files. Python deals with this with the “import” statement.

The 2 most commonly used formats are:

```
import file
```

and

```
from file imort function
```

Let’s try it (this is the all in one directory version):

```
1:
2: import jane
3:
4: jane.janefunc()
5:
```

```
1:
2: def janefunc():
3:     print ( "jane" )
4:
```

Multiple Directories Version:

```
1:
2: imort x.bob
3:
4: x.bob.bobfunc()
5:
```

```
1:
2: def bobfunc():
3:     print ( "bob" )
4:
```

or we can just import a single function

```
1:
2: from x.bob import bobfunc
3:
4: bobfunc()
5:
```

Copyright