Project 2 FAQ

Please read this FAQ before posting questions on piazza.

User Input Questions

Q1: Do we assume that the user will always enter the correct type for each of the variables, or should we account for type errors as well?

A2: Yes, you don't have to worry about invalid types in the purchase amount input if it is integer vs. float. However, an invalid type (input) in the month will cause an error. For example:

```
State Name: Texas
Purchase amount: 7
Provide the month: 10
Provide the day: 1
Provide the year: 2022
Invalid month!
```

Of course, a non-numeric input for the amount, day, and year will cause an error.

```
State Name: Texas
Purchase amount: five
Provide the month: October
Provide the day: 1
Provide the year: 2022
Invalid amount!
```

```
State Name: Texas
Purchase amount: 45.6
Provide the month: May
Provide the day: ?
Provide the year: 2022
Invalid day!
```

```
State Name: Texas
Purchase amount: 56.0
Provide the month: May
Provide the day: 5
Provide the year: This
Invalid year!
```

You need not worry about integer vs. float for numerical inputs for amount, day, and year. The following example is a valid case:

```
State Name: Texas
Purchase amount: 78
Provide the month: May
Provide the day: 1.0
Provide the year: 2022.0
Please pay a total of $84.39
```

Q2) The professor said that inputting 1.0 as the day is acceptable, but if the user inputs a number like 1.6 as the day, would this be allowed or should it print "Invalid day"?

A2) We will not have such a test case.

Q3) Can the user input a purchase amount that is more than 2 digits after the decimal? For instance, can they input a value like 35.6879 for the purchase amount or should this give them an error?

A3) Input of an amount more than 2 decimal places is fine. It is valid. You can test it out with the demo program provided.

Q4) Do we have to check for the user inputting a space in between numbers for the purchase, day, and year? If we are using cin, it takes in only the first number inputted before the whitespace, so can we assume that the user won't input multiple different numbers separated by white spaces for the purchase amount, day, or year?

A4) We will not have such a test case for grading evaluation.

Q5) If the user inputs an extremely long number of more than 15 digits for their purchase amount, should we treat anything at that number with an error message, attempt to fix it, or just ignore it? As it stands, when a user does that, it calculates the purchase nearly correctly but with a few random digits added or subtracted (some garbage, I guess).

So what should we do about that?

A5) Different compilers and systems produce the differences. We will not have such a test case (unreasonable amount) for grading.

Q6) Are we supposed to handle cases where the user inputs two numbers separated by a space as seen in the Purchase Amount?

```
State Name: Oregon
Purchase amount: 22 3
Provide the month: May
Provide the day: 12
Provide the year: 2022
Please pay a total of $22.00
```

A6) We will not have such test cases.

Q7) I'm using getline() for the state and cin for the other inputs, but then when a user inputs something like "May may" for month or "50 50" for amount, the program runs and just cins the first input which is valid. Should I use getline() for all of the inputs to prevent this from happening, or this what's supposed to happen?

A7) We will not have such test cases.

Other Questions

Q8) What information exactly are we supposed to use from the string handout? I was assuming we had to use cin.ignore() at some point but my code seems to be fine without it, should we be including that after every cin anyways?

A8) It depends on how you design your code. You may or may not need anything from the string info file.

Q9) One of the tax values was listed as negative (I think -0.03), was this a mistake or is that like a benefit that we should add to the total amount?

A9) No error. Some cities decided to have a negative local tax rate to offset the state tax rate.

Q10) Around how many test cases would you recommend we include in our report?

A10) It is for you to decide. You want to cover both valid and invalid test cases. The exercise for you is to think about the coverage you need.

Q11) Are we supposed to have an end line for all of the output statements, such as the error messages "Invalid month!", "Invalid day," etc, and then also for the "Please pay a total of $" since the specifications don't clearly show whether or not we are supposed to have a "\n" or endl.

A11) Yes, but given that there is only one output line; therefore, either way is acceptable.