

CS118

Problem Solving with conditions

Solving a problem that has "conditions" means that there are decisions that must be made by the solution. "Should I run these tasks?" is the decision that is always made, and the answer for that question depends on the data available to the solution.

Some commonly-seen examples are:

- Which option from a menu did the user choose?
Based upon the option, different tasks will need to be performed
- Did the user give an appropriate value?
If s/he did not, an error message can be produced. Obviously, we would not want that message otherwise.
- Is a computed value in a certain range?
If so, perhaps something else needs to be computed.

In the Standard Format, when a decision must be made, a TEST task is used. For example:

```
T03: TEST: If value less than 0, T04-T05
      T04: OUTPUT error message using value
      T05: DEFINE: add 1 to "errcount"
```

T03 is the decision to be made for this problem: "Should we handle this as an error situation? "

The task numbers (T04-T05) listed at the end of T03 are the **action tasks** that are to be executed if the condition for the TEST is true. There must be at least one, and there may be as many as you wish. Only those tasks listed at the end will be executed as a result of the condition. **These task numbers are always sequential and must immediately follow the TEST task.** The content of these tasks can be any legitimate tasks, including more TESTS.

Remember: You are providing instructions for what the The Solution is going to do. **The Solution will do nothing except what you tell it.** The Solution only has the ability to use The Tools. All other capabilities must be built from The Tools.

Common error: Students will attempt to perform tasks which seem obvious (as humans) but for which there is no logical sequence in the Solution:

```
T01: DEFINE "value_for_001" as 'abc'
T02: DEFINE "value_for_002" as 'def'
T03: DEFINE "value_for_003" as 'ghi'
T04: INPUT: id_number
T05: TEST: If id_number is 001, 002, or 003, T06
      T06: OUTPUT value_for_001, value_for_002, or value_for_003 as appropriate
```

For the erroneous solution above humans readily make the leap from T05 to whichever value should be output. But there is no means for the solution to make that determination. A more appropriate solution:

```
T01: DEFINE "value_for_001" as 'abc'
T02: DEFINE "value_for_002" as 'def'
T03: DEFINE "value_for_003" as 'ghi'
T04: INPUT: id_number
T05: TEST: If id_number is 001, T06
      T06: OUTPUT value_for_001
T07: TEST: If id_number is 002, T08
      T08: OUTPUT value_for_002
T09: TEST: If id_number is 003, T10
      T10: OUTPUT value_for_003
```

On the following page is an old exam as an example problem with its solution on the next page. Try to develop it on your own and refer to the solution only when you get stuck. Your assignment to turn in is on the last page.

Example problem

CS118 Program Oil Change 2



Your new boss has decided to make use of your computer programming skills to create a program for customer self-service. As part of your development activity, your program will allow an *employee* to use the program to construct a menu for the user. Have the program ask for two make/model combinations with recommended oil change period in miles, and the price (each model could have a different price).

Use the inputs to show a menu to the user (see the example run below). Find out how many miles have elapsed since the last oil change and if the actual mileage since the last change is within 10% of the recommended mileage for that model (either above or below), provide the user a 20% discount. Tax is 6.5% on the discount-adjusted price. Provide a final quote that appears as the example output shown in the sample run:

[HINT: Hardcode the discount and save that determination for last.]

```
Employee use only:
```

```
What is the make and model? Chevy Corvette
What mileage is recommended between oil changes? 5000
What is the price for this make and model? 100.95
```

```
What is the make and model? Infiniti Q45
What mileage is recommended between oil changes? 4000
What is the price for this make and model? 35.94
```

```
1. Chevy Corvette           5000      100.95
2. Infiniti Q45            4000       35.94
Which model (1/2): 2
```

```
How many miles since the last oil change? 4005
```

Make/Model	Recommended change	Actual change		
Infiniti Q45	4000	4005	\$	35.94
	Mileage Discount:		\$	7.19
	Tax (6.5%):		\$	1.87
	Final quote:		\$	30.62

Remember - this is a sample run

Oil Change SOLUTION

Note 0: Before you start

Don't add task numbers until you have all tasks in place. This will prevent you having to renumber them later.

T01: INPUT: car make and model as "car01"
T02: INPUT: recommended mileage as "recommend01"
T03: INPUT: price of oil change as "price01"
T04: INPUT: car make and model as "car02"
T05: INPUT: recommended mileage as "recommend02"
T06: INPUT: price of oil change as "price02"

Note 1: T01-T06

Use of variable names of the same form with numbers makes it easy to keep track of related information.

T07: OUTPUT: Display menu for user using **all input data**

Note 2: T07

Summarizing information is fine as long as it is unambiguous.

T08: INPUT: Collect user's menu choice as "choice"
T09: INPUT: Miles since last change as "miles"

T10: **DEFINE: "discount" as 0**

Note 3: T10

By starting "discount" with the value 0 we prevent having to put in another TEST and DEFINE after T13 and T19.

T11: TEST: if choice = 1, **T12-T16**

T12: TEST: if **miles within 10% of recommend01**, T13
T13: DEFINE: "discount" as 20% of price01

Note 4: T11

Common student error: Every TEST must specify which tasks will execute if the condition is true. If the condition is false, those tasks will be skipped.

T14: DEFINE: "tax" as $0.065 * (\text{price01} - \text{discount})$
T15: DEFINE: "final" as $\text{price01} - \text{discount} + \text{tax}$

Note 5: T12

Conditions and descriptions don't have to be mathematics if English is easier. Be sure the English is accurate, though!

T16: OUTPUT: table of final quote using car01, recommend01, price01, discount, tax, final

T17: **TEST: if choice = 2, T18-T22**

Note 6: T17

TEST has no ELSE option, so you must present another TEST task here.

T18: TEST: if miles within 10% of recommend02, T19
T19: DEFINE: "discount" as 20% of price02

T20: DEFINE: "tax" as $0.065 * (\text{price02} - \text{discount})$
T21: DEFINE: "final" as $\text{price02} - \text{discount} + \text{tax}$

T22: OUTPUT: table of final quote using car02, recommend02, price02, discount, tax, final

Note 7

This solution may not be exactly as you might prefer. For example, there is no handling of potential error, such as would occur if the user didn't choose one of the two possible menu choices. Note that the problem does not require error handling so you should not include it – follow the assignment.

Exercise

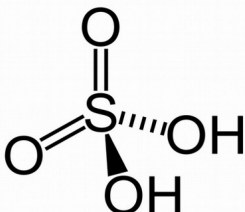
Prepare a solution in the Simplified Standard Format for this problem:

CS118 Programming Acid Concentration

As any beginning chemistry student can tell you, the pH of a solution is a measure of the acidity of that solution. Specifically the pH is a measure of the hydrogen ion H^+ concentration:

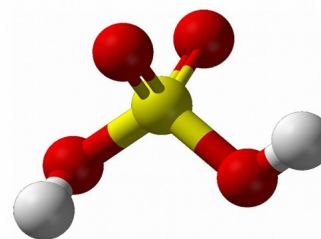
$$pH = -\log_{10}[H^+]$$

where $[H^+]$ indicates the concentration of the hydrogen ion in moles per liter of solution.



The concentration ("molarity") of a solution is measured in moles of solute per liter of solution ($M = \text{moles/liter}$). A mole is a convenient abbreviation for a large quantity of atoms or compounds (specified as Avogadro's Number = 6.022×10^{23}). One mole of any compound is made up of a proportionate number of moles of its elemental components – thus one mole of H_2SO_4 (sulfuric acid) contains two moles of hydrogen atoms, one mole of sulfur atoms, and four moles of oxygen atoms. Using atomic weights, we can convert easily between moles and grams. One mole of sulfuric acid, for example, weighs about 98 grams: one mole of hydrogen weighs 1 gram, one mole of sulfur weighs 32 grams, and oxygen weighs about 16 grams per mole. (BTW, water has the molecular formula H_2O and a density of 1000g/liter – you can figure out the molecular weight yourself.)

Polyprotic acids (more than one acidic hydrogen) do not completely dissociate the hydrogen atoms from the compound. For our purposes, though, we will assume that they do. Provide a computer program that will request the user provide the number of hydrogen atoms first – if that number is not an integer or is not between one and three, produce an error message indicating the specific error and have the program terminate. In all other cases, collect from the user the pH of the acid solution, the name of the acid in the solution, and the molecular weight of the acid. Compute the mass of acid in 100ml of solution for any acid with one to three hydrogen atoms and display the information in a table as shown in the example below.



```
How many hydrogen atoms in the solute? 2
Name of acid in solution? Sulfuric Acid
Molecular weight of acid? 98
pH of acid solution? 2.99
```

Name	pH	MW	atoms H	g/100ml
Sulfuric Acid	2.99	98	2.00	0.0050

```
How many hydrogen atoms in the solute? 1.3
Error! There must be an integer number of hydrogen atoms!
```

```
How many hydrogen atoms in the solute? 4
Error! What has that many acidic hydrogens?!
```

```
How many hydrogen atoms in the solute? 2
Name of acid in solution? Unobtanatic Acid
Molecular weight of acid? 11802
pH of acid solution? 0.5
```

Name	pH	MW	atoms H	g/100ml
Unobtanatic Acid	0.50	11802	2.00	186.6060

Approach:

· In two of the sample runs we see four inputs being provided and one table of output. However in the other two sample runs we see only one input and one error message. This suggests that three of the inputs may not be executed – the TEST tool allows us to decide if we are going to execute a task. Similarly, in certain circumstances the table may not be produced, and in other circumstances the error messages may not be produced. So TEST tools will permit use to decide when each of these tasks should be executed.

· The tables require computed values, but the problem says to "have the program terminate" if the input is faulty. This means that there will be no computation of those table values if the inputs are faulty (and no table will be produced). Once again, the TEST tool allows us to decide if we are going to compute those values.