EECS3311 Fall 2020 Questions for Lectures: Week 1

There are three sections of this document:

- **Slides**
- **Lecture/Tutorial Videos**
- **Confusion**

Please choose the most applicable sections for your questions.

Slides (Topic? Slide number? Bullet point?)

1. Lecture Notes Week 1, Slide 27, we mentioned how exceptions (logical negations of preconditions) were not good enough to take care of the case where balance == amount, but wouldn’t that also be true for the constructor? Ie. the precondition for the constructor wouldn’t catch if the balance = 0, therefore violating Requirement 1?

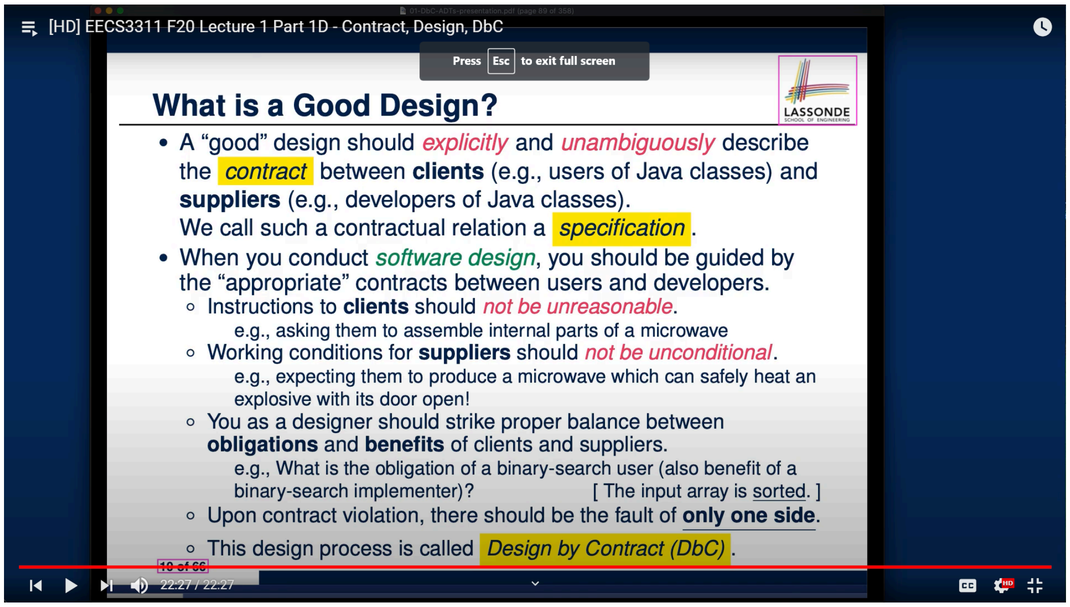
   Jackie’s response: Indeed. The exceptional condition in Line 5 should be:
   balance <= 0.
   I think it was not mentioned explicitly because I meant to just focus on the exceptional condition at Line 11.

Lecture/Tutorial Videos (Which part? Timings?)

2. This is Question 1
   "Part 2C - Bank Accounts V2 (weak precondition)" at 17:55
   is not an acceptable balance because requirement #1 says that the integer balance is always positive.
   But doesn't BalanceNegativeException (lines 5-6) violate that requirement? The client could create an AccountV2 object with input balance being 0, because the error condition (balance < 0) is false, so line 7 proceeds and the object is initialized.

   Jackie's response: Indeed, spot on! The exceptional condition in Line 5 should be: **balance <= 0**.
   I think it was not mentioned explicitly because I meant to just focus on the exceptional condition at Line 11.

3. From Lecture 1 Part D Slide 10/66, you said that there should be **exactly one** violation of the contract (by either supplier or client) and that it's better than having none or both violate the contract… I understand why it's better than both violate, but I don't understand why is it better than none violating the contract (in other words, shouldn't the best case be **no** violations of the contract by **either** supplier nor client).


[HD] EECS3311 F20 Lecture 1 Part 1D - Contract, Design, DbC
Press Esc to exit full screen
LASSONDE
10 of 66
22:27 / 22:27

What is a Good Design?

- A "good" design should *explicitly* and *unambiguously* describe the *contract* between **clients** (e.g., users of Java classes) and **suppliers** (e.g., developers of Java classes).
  We call such a contractual relation a *specification*.
- When you conduct *software design*, you should be guided by the "appropriate" contracts between users and developers.
  - Instructions to **clients** should *not be unreasonable*.
    e.g., asking them to assemble internal parts of a microwave
  - Working conditions for **suppliers** should *not be unconditional*.
    e.g., expecting them to produce a microwave which can safely heat an explosive with its door open!
  - You as a designer should strike proper balance between **obligations** and **benefits** of clients and suppliers.
    e.g., What is the obligation of a binary-search user (also benefit of a binary-search implementer)? [The input array is sorted.]
  - Upon contract violation, there should be the fault of **only one side**.
  - This design process is called *Design by Contract (DbC)*.

Jackie's response: Good question. Your question may be boiled down to: "Is having no contract violation the most ideal scenario we'd hope for?" Not quite. Say the implementation is faulty, but the postcondition is unable to catch it by a violation, then the supplier's obligation has not been properly specified (which is unfair to the client). We will talk about examples during the Q&A.

4. Have you posted a link for lab 0? Where can I find it?

    Jackie's response: I'm sure you've found it from the eClass already.

5. When is lab 0 due? The syllabus says week 3 but the calendar says week 2.

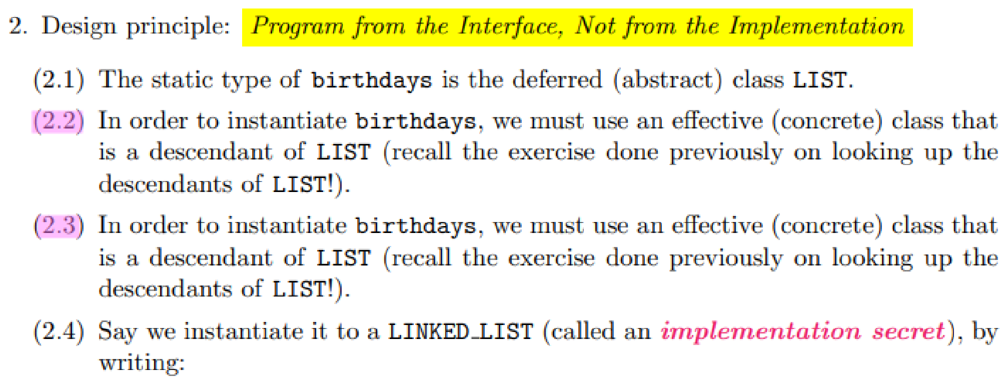
    Jackie's response: for lab/project deadlines, always refer to their instructions.

contract

6. What exactly does the "contract" part in design by contract mean? Is it referring to the boolean expressions stated in the precondition, postcondition, and class invariant? So when there is a contract violation, the violation refers to a condition where one of those boolean expressions is false?

Jackie's response: I like this question! Yes, contract refers to the Boolean expressions specified under **require**, **ensure**, and **invariant**; where there is a contract violation, one of these Boolean conditions evaluates to false. However, philosophically, think of contract as adding an extra layer of declaration properties (using predicates), against which your implementation/code is checked against. I will explain this in detail in class.

7. For the video 17 tutorial notes, 2.2 and 2.3 (highlighted in pink) are stating the same thing. Was there an important point that you meant to make for one of them instead, or is the double entry simply a typo?



2. Design principle: *Program from the Interface, Not from the Implementation*

    (2.1) The static type of `birthdays` is the deferred (abstract) class `LIST`.

    (2.2) In order to instantiate `birthdays`, we must use an effective (concrete) class that is a descendant of `LIST` (recall the exercise done previously on looking up the descendants of `LIST`!).

    (2.3) In order to instantiate `birthdays`, we must use an effective (concrete) class that is a descendant of `LIST` (recall the exercise done previously on looking up the descendants of `LIST`!).

    (2.4) Say we instantiate it to a `LINKED_LIST` (called an *implementation secret*), by writing:

    Jackie's response: Apparently 2.2 and 2.3 are a duplicate of each other. I updated the PDF. Nonetheless, think of this as an emphasis on how important this design principle is!

8. In Tutorial Part 5: Why does the first incidence of the incorrect query with the assertion result in an assertion violation error but the second incidence results in a postcondition violation error? Both incidents violate the postcondition so shouldn't both of them result in a postcondition violation error if the postcondition is checked before the assertion is checked?

    Jackie's response: I will clarify this in the Q&A session.

9. In Lecture 1 Part 2A (13:52) , you mentioned that the supplier and client should be blamed if the amount value is negative. My question is why is it still client fault since the client followed the supplier information(manual) who did not provide that info precisely ?
10. Is it bad design to have if statements in the post conditions or is it acceptable?

Confusion (Which concept?)

11. This is Question 1
    Bank Account V4 (faulty implementation), 6:18
    So when there's a contract violation, only one side should be blamed. But in this example, you said that it is not the case. Is it because both supplier and client are to be blamed, or neither?

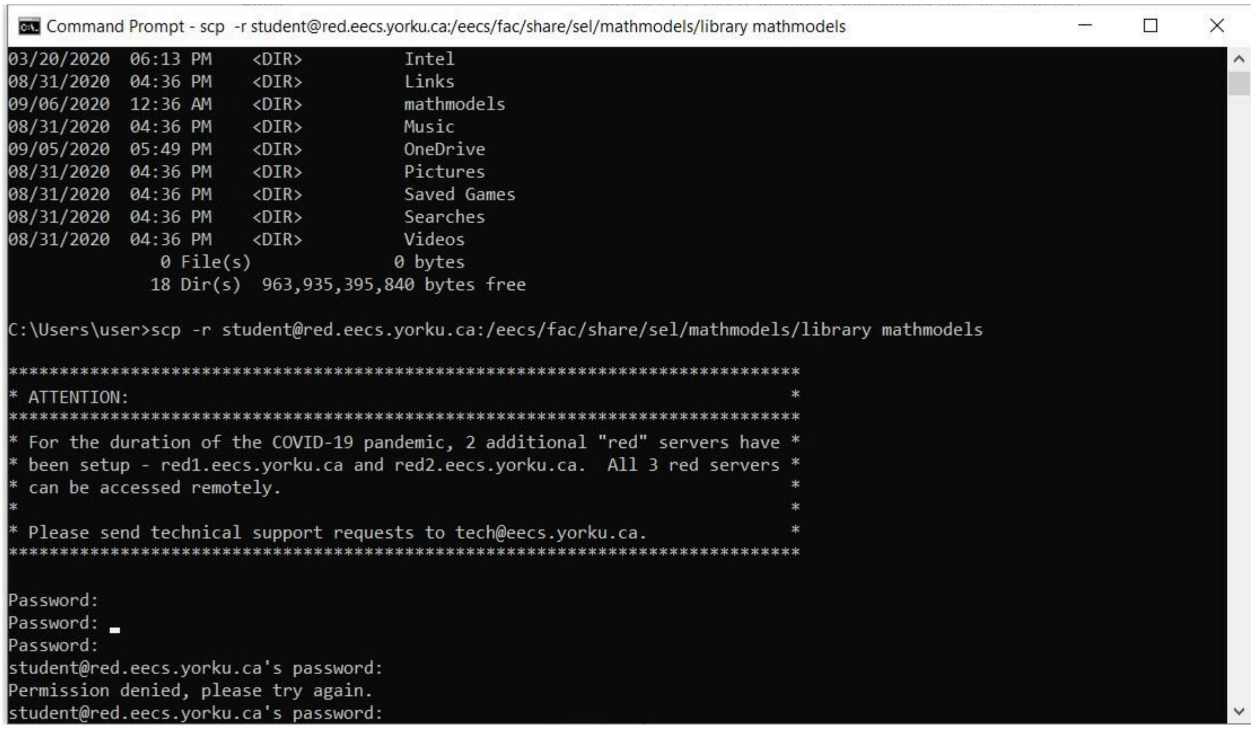
    Jackie's response: In this case, in the absence of a proper postcondition, is there any contract violation? No, because the invariant (balance > 0) can still be specified by the faulty implementation. Seemly, there's no contract violation, so neither the client nor the supplier would be blamed. But, the resulting balance (150, which should be 50) is not as expected. Something goes wrong, but there's no contract violation, so the design is flawed.

    It's clear that the supplier is the only one at fault because they have a faulty implementation. The client couldn't have known that the implementation was faulty, and they've satisfied the precondition anyways (Jeremy has a starting balance of 100 and withdrew 50, both valid). The client satisfied their obligations but the supplier did not.

    Jackie's response: It is true that it's the supplier's fault that their internal/private implementation is wrong, but the law/contract does not enforce it, so it's still considered as legitimate. I will illustrate this point again.

12. Referring to http://seldoc.eecs.yorku.ca/doku.php/eiffel/faq/mathmodels, on a Windows 10 machine, instruction unclear on what to do. Download the zip from the link provided? Use the scp command and grab it off the red server?

Jackie's response: Set the environment variable pointing to the path where you store the *mathmodels* folder.



Command Prompt - scp -r student@red.eecs.yorku.ca:/eecs/fac/share/sel/mathmodels/library mathmodels

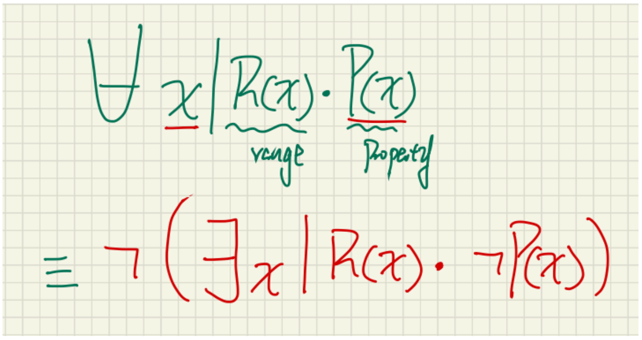
```
03/20/2020  06:13 PM    <DIR>          Intel
08/31/2020  04:36 PM    <DIR>          Links
09/06/2020  12:36 AM    <DIR>          mathmodels
08/31/2020  04:36 PM    <DIR>          Music
09/05/2020  05:49 PM    <DIR>          OneDrive
08/31/2020  04:36 PM    <DIR>          Pictures
08/31/2020  04:36 PM    <DIR>          Saved Games
08/31/2020  04:36 PM    <DIR>          Searches
08/31/2020  04:36 PM    <DIR>          Videos
               0 File(s)              0 bytes
              18 Dir(s)  963,935,395,840 bytes free

C:\Users\user>scp -r student@red.eecs.yorku.ca:/eecs/fac/share/sel/mathmodels/library mathmodels

******************************************************************************
* ATTENTION:                                                                 *
******************************************************************************
* For the duration of the COVID-19 pandemic, 2 additional "red" servers have *
* been setup - red1.eecs.yorku.ca and red2.eecs.yorku.ca.  All 3 red servers *
* can be accessed remotely.                                                  *
*                                                                            *
* Please send technical support requests to tech@eecs.yorku.ca.             *
******************************************************************************

Password:
Password:
Password:
student@red.eecs.yorku.ca's password:
Permission denied, please try again.
student@red.eecs.yorku.ca's password:
```

Tried to use blank password, but “permission denied”

13. Dear prof could you mention the section of two textbooks for the required reading part each week please? Thank you.

Jackie’s response: Actually that’s what I did! If you look at the title page of the slides, you’ll see them.

14. How exactly do you read this dot symbol? Is it just another symbol for conjunction? And if so, shouldn’t R(x) also be negated, and shouldn’t there be a disjunction between not R(x) and not P(x)? There must’ve been some double negation to get from the expression in green to the expression in red. (From Tutorial Part 20)



$$\forall x \mid \underbrace{R(x)}_{\text{range}} \cdot \underbrace{P(x)}_{\text{Property}}$$

$$\equiv \neg(\exists x \mid R(x) \cdot \neg P(x))$$

Jackie’s response: I will clarify this in the Q&A session!

Lecture/Tutorial Videos

Slide 7 Video 4
1- Professor how the assumption of instructions being followed is the benefit for the supplier?

Jackie's response: I will address this in the Q&A session. But just to recall the example, isn't the fact that a microwave manufacturer **doesn't need to produce a microwave which can heat a lunchbox with its power off and door open** a benefit for them?

2- why the violation of contract should only be from one side? Why not both the sides and what will happen if there is no violation? I am kind of confused about this part.

Jackie's response: This question overlaps with what your fellow students were asking. I will address it in the Q&A session. In short, having no contract violation isn't necessarily a good thing (e.g., you can simply just put both pre-condition and postcondition to be **true**). In that case, you will not have any contract violation, but your system may just not behave as expected or crashed.

3. Given violation of precondition will not execute the feature call. What is actually happening inside the cache values during the run time encountering postcondition or class invariant violation? How does the compiler deal with the new assigned values?

4. In EECS3311 F20 Lecture 1 Part 2D - Bank Accounts V3 (class invariant) is it possible to have different invariants for different methods of the same class?