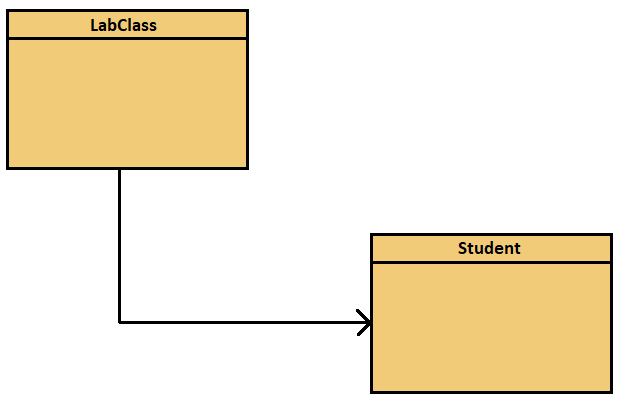
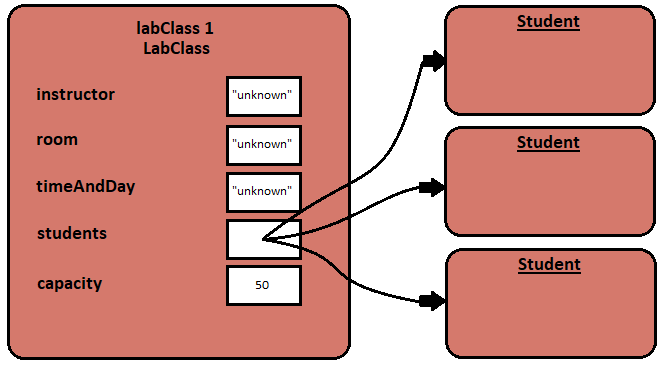
Chapter 3 Exercises

* 3.1
  + Class Diagram



* + Object Diagram



* + The class diagram shows the static view, meaning it depicts what we have at the time of writing the program. It just shows the two classes LabClass and Student. It shows that the class LabClass makes use of the class Student, meaning the class Student is mentioned in the source code for LabClass. On the other hand, the object diagram shows the dynamic view, meaning it shows the situation at runtime. It shows that the three objects of Student we created an LabClass object.
* 3.2
  + Class diagrams can only change when writing or editing the program. It can only be changed by editing the source code of the program.
* 3.3
  + The object diagram can change during the execution of the program (runtime). It changes when new objects are created or objects are destroyed in the program.
* 3.4
  + private Instructor tutor;
* 3.5
  + When you call the increment method, the terminal window shows:

hours.increment();

* 3.6
  + When you call the increment method, the terminal window shows:

minutes.increment();

* + The increment method doesn’t indicate when the rollover limit is reached, so to check whether or not you have reached this limit, you should call the getValue method and check to see what number it is at. Once you see that it is at 59, you know that it will roll over if you increment it by 1, so you know you have to increment the hours object too.
* 3.7
  + I created a NumberDisplay object in the Code Pad and called its getValue, setValue, and increment methods in the Code Pad.
* 3.8
  + The error message is:

Error: non-static method getValue() cannot be referenced from a static context

* 3.9
  + The error message is:

Error: ‘.class’ expected

* + The error made is that there should not be the word int in the actual parameter. It should be:

nd.setValue(5);

* 3.10
  + Nothing will happen if the setValue method is called with an illegal value.
  + This is not a good solution because the user will not understand why the program isn’t working.
  + A better solution is to print a message like “Please enter a value greater than zero and one less than the limit” so that the user knows what to do to make the program work correctly.
* 3.11
  + If you replaces the “>=” operator with the “>” operator, then the setValue method will also not accept 0 as an actual parameter.
* 3.12
  + If you replaced the && operator with the || operator, the program will run as long as one of the conditions is met, meaning that you could put in a negative value or a value greater than the limit.
* 3.13
  + The following are true
    - ! false
    - (34 != 33) && ! false
* 3.14
  + ( a && b) || (!a && !b)
* 3.15
  + (a && !b) || (!a && b)
* 3.16
  + a = b
* 3.17
  + The getDisplayValue method does work correctly in all circumstances as it will always display the value. However, in terms of the clock, it wouldn’t make sense because it doesn’t check if the number being displayed is a 2 digit number. Therefore, if you create a number display with limit 800, it will display a 3 digit number.
* 3.18
  + No, it doesn’t matter whether the string is first or second in the expression. As long as one of the values is a string, the other operand will always be converted to a string.
* 3.19
  + 9 + 3 + “cat”
    - Prediction: “12cat”
    - Test: “12cat”
    - No this did not surprise me
    - Because the program reads the expression from left to right and 9+3 came first, that operation was carried out first and therefore 12 was added to “cat”, and produced “12cat”.
  + “cat” + 3 + 9
    - Prediction: “cat39”
    - Test: “cat39”
    - No this did not surprise me
    - Because the program reads the expression from left to right and “cat”+3 came first, that operation was carried out first and 3 was converted to a string and added on the cat, therefore resulting in “cat39”.
* 3.20
  + The modulo operator returns the remainder of the two numbers in the expression. It also works with floating point values.
* 3.21
  + The result of (8 % 3) is:

2