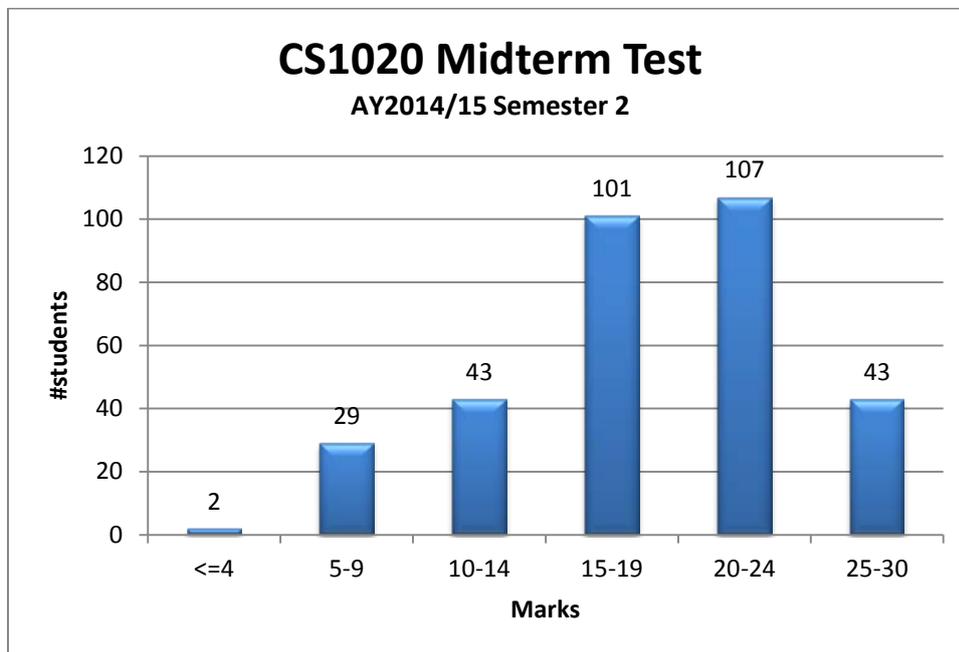


Comments on CS1020 Midterm Test (AY2014/15 Semester 2)

325 students sat for the test on 4 March. The mean is 18.45 out of 30 marks, or 61.5%.

I set this test paper with the aim of getting a mean between 63% and 65%, so I have missed it by a small margin.

The distribution of the marks, in brackets of 5 marks, is shown in the chart below.



I would think the paper is of reasonable level of difficulty – neither too easy nor too hard. I would urge the 74 students who scored below 15 marks, and even more so for those who scored below 10 marks, to work really hard and revise the materials for the final exam, which has about 30% on materials before the recess and 70% on materials after the recess.

The answers are on the CS1020 website under “Term Tests”:

http://www.comp.nus.edu.sg/~cs1020/3_ca/termtests.html

You may check your marks on the IVLE gradebook. We will return your answer sheets to you at your next tutorial.

Appeal: Some of you wrote in really tiny handwriting, which makes it really challenging for this old man. Please, in your final exam, write **BIGGER!**

2. Discussion on MCQs

The table below shows the percentage of students who chose the correct answers, and chose the most popular wrong answers:

	Q1	Q2	Q3	Q4	Q5
%students who chose the correct answer	C (72.0%)	E (40.3%) Hardest	B (50.2%)	C (72.9%) Easiest	D (40.9%)
%students who chose the most popular wrong answer	D (15.4%)	D (43.7%)	E (20.6%)	D (18.8%)	C (13.8%)

Q2 is the hardest MCQ. The answer is (E) Compilation Error, but most students answered (D) MismatchedMethodException. There is no such exception by the way.

3. Discussion on Questions 6 – 11

The table below shows the average marks for each question, normalised as percentages.

Q6	Q7	Q8	Q9	Q10	Q11
44.5%	71.5%	40.3% Hardest	52.7%	66.3%	72.7 Easiest

I would like to comment on a few questions.

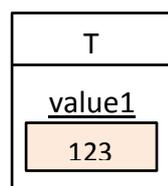
Q8:

This surprised me as the hardest question as I thought the concepts are basic: class attribute versus instance attribute.

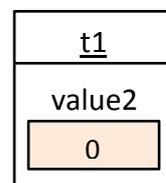
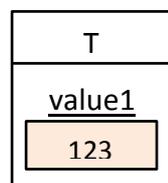
Class attributes are shared by all instances in that class. There is only one copy, and all instances can access that copy. Instance attributes are stored in individual instances.

Note that while a class method cannot access instance attributes, an instance method may access class attributes.

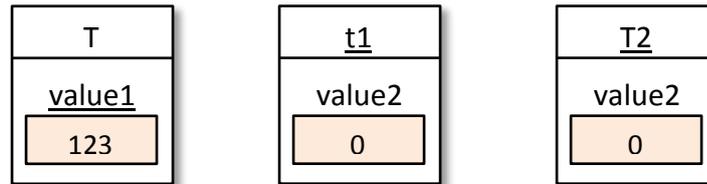
Let's trace the code. In the beginning:



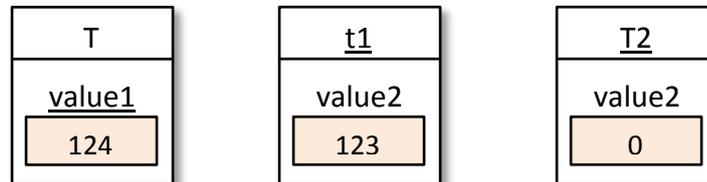
```
T t1 = new T();
```



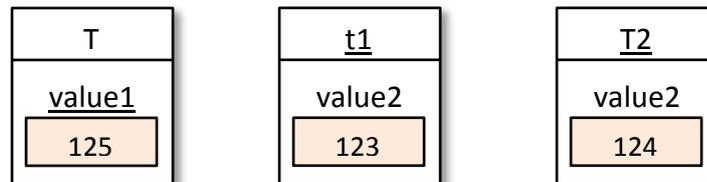
```
T t2 = new T();
```



```
t1.increment();
```



```
t2.increment();
```



Q9.

This is a rather straight-forward question, hence I expected good performance. However, many students did not use the right String methods, or mixed up characters with Strings. The String API was not given in the paper because String is a very commonly used class so I expect students to be familiar with its methods.

There are many alternative answers. Two are shown in the answers provided:

A one-liner:

```
word2 = word2.replaceFirst(word1.substring(i,i+1),"");
```

or:

```
String letter = word1.substring(i, i+1);
if (word2.contains(letter))
    word2 = word2.replaceFirst(letter, "");
else
    return false;
```

The second solution, though longer, is more efficient as it returns false the moment word2 is found not to contain the inspected letter in word1.

If you have forgotten about the **contains()** or the **replaceFirst()** method, or don't know that they exist, there are alternative solutions still. For example, you may use **indexOf()** and **charAt()**, as follows:

```
int pos = word2.indexOf(word1.charAt(i));
if (pos >= 0)
    word2 = word2.substring(0,pos) + word2.substring(pos+1);
else
    return false;
```

Q10.

I am quite disappointed that a number of students didn't use **this** for parts (a) and (b) to write the overloaded constructors. This is a very basic feature in Java, and you should have done it in practice exercise 10 (Ball with centre) and 11 (Fraction).

For part (c), it is a simple matter of finding out the length and width of the triangle, by using **Math.min()** and **Math.max()**.

Many students wrote something like this:

```
return 1/2 * ...;
```

This will evaluate the expression to zero! I didn't deduct mark (but I think I should) because I was focussing on more complex mistakes.

Some students used rather complicated formulas, such as Heron's formula, which is unnecessary here. Some students used methods such as **hypot()** and **sqrt()** which may result in inaccurate answers. I did not deduct marks but just commented that these are not necessary.

Q11.

The first line in the answer for part (a) is:

```
triangles = new ArrayList<Triangle>();
```

Some students omitted the above line. In this case the attribute **triangles** is not constructed, and hence the later statement `triangles.add(...)` will throw a `NullPointerException` error.

Many students wrote one of the following instead:

```
ArrayList<Triangle> triangles = new ArrayList<Triangle>();
ArrayList<Triangle> list = new ArrayList<Triangle>();
```

They are incorrect because the first statement will create a *local reference* **triangles**, not the attribute **triangles** declared at the beginning of class **TestTriangle**. In both cases, the attribute **triangles** is not used.

All the best for the subsequent sit-in labs and your final exam!

Aaron Tan
9 March 2015