LAB 1 Data Structures
CSI 201
Spring 2012

DUE: On Blackboard, Sunday, January 29 at 11:59PM. Type or upload writing in English plus Java expressions following instructions at the end. Upload

The technology goal of this lab is to make sure you can run DrJava in the lab. The learning goals are:

1. Work as a human 1890s-1940s computer who uses Euclid’s remainder algorithm with paper and pencil, to get a clear idea of what that means.
2. Use the DrJava interactions window to command Java arithmetic calculations to do the remainder-finding.
3. Use (1) Java declarations, (2) assignment statements and (3) `System.out.println()` method calls to make the Java system keep track of the numbers as a backup for your computer person work on paper.
4. Start, like a computer scientist, to recognize and describe patterns in computed answers.

Prelab:

1. Review and practice the subtracting onto Pink and Yellow tickets done in the lecture. When we started with 57 on the Pink ticket and 39 on the Yellow, we ended up with a 3 and a 0. The answer was 3.

Reminder and makeup explanation: Repeatedly, look at the numbers and see which ticket has the bigger number. Subtract the smaller number from bigger number and replace the bigger number on the ticket after crossing it out. In case the numbers are equal, just pick one of them to subtract, so the other one gets replaced by 0. When one numbers is 0, stop and the answer is the other number.

2. Make sure you have and know or have written down both your UAlbany NetId which is in your UA email address AND your MyUAlbany (secret) password.

InLab:

Part 1: Repeat the lecture computation started with Pink containing 52 (instead of 57) and Yellow containing (as before) 39. However, instead of using tickets, illustrate what the computer (male or female person) sees by making a comic strip or movie frame-by-frame.

Answer briefly in English: What happens after one of the tickets first gets the final answer number written on it? In the (52, 39) computation the computer gets the answer 13, assuming she or he didn’t make a mistake. (You can finish after the lab.)

Part 2: Start DrJava!

1. Follow the instructions posted, given out and explained. Briefly, log in and command: `/usr/local/depts/cs/geds/drjava.jar`

2. Make the bottom sub-window big: Drag its top boundary up. Then click the tab to see the Interactions sub-window.

3. Test the Interactions window by typing your first Java method call: `System.out.println(2012);` and press enter. If the computer doesn’t print 2012, ASK FOR ASSISTANCE!

Part 3: Work as a computer (person, male or female) while using a computer (electronic machine) as a helper.

You will do computations similar to those from lecture and practiced above, but different because you will calculate remainders (from division) instead of doing subtraction.

The electronic computer should help you in two ways: (1) Like before, it will store the numbers on the Yellow and Pink tickets electronically. (2) Instead of you manually computing the remainder from dividing two integer (whole) numbers, you will write Java statements that make the electronic computer do that calculation upon the numbers written on the tickets. Each statement will also overwrite the number written on the correct ticket with the remainder from division.
Technical (which pedal, brake or gas?) fact you need: Just like the expression $103 - 20$ commands the computer to calculate the result $83$ by subtracting, the expression

$$103 \% 20$$

commands the computer calculate out the remainder when dividing $103$ by $20$. It should calculate $3$.

Of course, suppose the numbers are written on tickets (which experts call “variables”) with names like Yellow and Pink. The statement that commands the computer to get the remainder when the number on the Yellow (ticket) is divided by the number on the Pink and then overwrite the number on the Yellow with the remainder is

$$\text{Yellow} = \text{Yellow} \% \text{Pink};$$

(An expert would say “value of variable (named) Yellow” instead of “number on the Yellow (ticket)”. If he/she is lazy, you’d hear “value of Yellow” or just “Yellow”.

That’s why beginners can’t understand it when experts talk to each other! The assignment operation (denoted by $=$) really does overwrite the current “value of the variable named Yellow” with a usually different value.

PostLab: Type or attach in the Blackboard Lab 01 assignment your answers to these questions.

Question 1: What did you find hardest about Lab 01? What might the professor go over in class to help with that?

Question 2: The Java statement $\text{Yellow} = \text{Yellow} \% \text{Pink}$; contains 6 tokens, the semicolon is the last of the 6. How many English words did it take on the lab assignment sheet to explain what that statement commands the computer to do? Copy 6 of those words to prove that you located the right words that the question is about.

Question 3: (a) Identify and explain a pattern that the subtracting kind of computation from the lecture seems to follow AFTER the number that is the answer is first written on one of the tickets.

(b) Identify and explain a pattern that the Lab 01 computation seems to follow AFTER the number that is the answer is first written on one of the tickets. That’s the computation with getting remainders from Java using the $\%$ operation.

Question 4: Explain how division of (positive) integers getting an integer quotient and remainder is really a kind of repeated subtraction. Include in your written explanation (a) What is the significance of the quotient? (b) What is the significance of the remainder? This writing might be easiest if you imagine a person who figures out the quotient and the remainder by repeatedly subtracting the divisor (the number you divide by). It might also help to do it on paper with small particular numbers like $17$ divided by $5$. Their quotient is 3 and the remainder is 2. The computation is $17 - 5$ is $12$, $12 - 5$ is $7$, $7 - 5$ is $2$ and it stops because $2 < 5$. Exploring examples on paper is what computer science seniors and graduate students do when they study much newer algorithms.