CS 200 Sections 02 & 04 Spring 2013

Week #9: Top 3 Lessons Learned

1. Computer read the code one by one! Therefore:

System.out.print(a++); // **first display a, then increment**
System.out.print(++a); // **first increment, then display incremented value**

Important! a stays incremented!

2. One more time, operator precedence is extremely important.

http://www.neiu.edu/~faporps/2013Spring/cs200/00OperatorPrecedenceTable.pdf

AND is always before OR

3. Value can be incremented or decremented inside of if ()

if(d == c--) // first the value not decremented is compared to d, then decremented.

M. Mardosz

What I learned on 3/7/13 was **before starting any flowchart you must read the question carefully and write down what is given and what its asking**.

If data is being entered from the user create a valid/invalid data decision structure.

When reading any question or problem, follow the direction given to you and write down the steps the computer will be running before coding the program.

When creating a valid/invalid decision structure problem, write down which one is valid and invalid. Example: if (Input years > 1852) { //valid System.out.print ("This is true"); } else { //invalid System.out.print ("This is false"); } let say that the input years are 1920s,1873s, and 1840s. 1920s and 1873s are valid data because they are greater than 1852. 1840 is invalid because its less than 1852.

P. Khuu

1. **When code tracing, you should first look at the variables that are declared and write them out vertically on the side, so that you can keep track and change the values as you need to.**
2. **When tracing, be sure to look carefully for semicolons followed by another command/statement on the same line.** An error can easily be made in stating that the statement is everything on the same line when there are actually two commands on the same line.
3. In addition to tracing, **remember to put a dot on the next line if you have System.out.println() to remind yourself that the next statement of output needs to go on the next line.**

T. Blanchard

Three things that I learned on 3/05/2013 are that **it is important to pay attention to the placement of the increment operator and the decrement operator. If they are placed in front of a number, you have to change that value before you continue with the expression. If they are placed after a number, you have to change that value after it has been used in the expression.**

Second, you have to pay close attention when you are code tracing. You have to keep track of the current value of a variable because the value of a variable can change as you proceed through the code.

Third, you also have to pay attention to the methods println and print. When you are code tracing, you have to pay attention to these methods so that you know if you are going to start on a new line or not.

J. Gomez

When tracing a problem, write the variables vertically so you can see clearly the values stored in each variables.

When you change a variable's value, cross it off in a way that lets you see what you have crossed off.

Always put your finger on the line you are working on!

When the if statement is true make sure you cross off the else statement that you don't need to do

Always write the value that is assigned to the variable when looking at an if/else statement

S. Malik

1.) **prefix operators ++, -- come before the variable and tells the computer to add/subtracts one to the variable's value, and if prompted to, show it on the output. At the other end of the spectrum we have the postfix operators that come AFTER the variable, it means the computer uses the original variable value FIRST in the statement, THEN it increases or decreases the value by 1.**

2.) **An if and an else statement can one process ONE statement when no {} are used to compound them.** When tracing on the final, make sure to see if the statements are in brackets for the if and else statements; if not then processed in the linear fashion and do the next statement.

3.) **Remember the order of precedence!! Also make sure you FOLLOW the direction of the operator's associativity (r to l or l to r), not doing so WILL lead to a wrong answer so always double check it.**
E. Herring

A few of the lessons learned were from tracing code:
1. **Take the variables and write them on their own line. This way, you can make the changes to the variables as you trace the code and it will be easier to read.**
Example:
A =1
B =2
etc.

2. **When tracing the code, write T or F above the logical comparisons so you can keep track of whether the evaluation is true or false.**

3. Read through each line to make sure you don't miss a new line character. You'll need to show that in the box provided.

E. Zacharias

1. **To solve a problem with code tracing, we go line by line to find the results.**
2. **If you want to include more than one statement in each branch, simply enclose the statements in braces { }. Several statements enclosed within braces are considered to be one larger statement. This is called compound statements.**

3. **Using = Instead of = = to Test for Equality.**
The operator = is the assignment operator. Although this symbol means equality in mathematics, it does not have this meaning in Java. If you write if (x = y) instead of if (x == y) to test whether x and y are equal, you will get a syntax error message.

D.Mirdadi