ICSI201 Spring 2012 Lab 10

Read this side first. HOW to do it is printed here. WHAT to do is printed on the reverse.

Make a fresh Lab10 directory/folder that MUST be under a directory/folder dedicated to CSI201. It MUST be named **`Lab10`** Do this QUICKLY: Get help from a neighbor or TA right away!

In this lab you will (a) practice keeping a version history like that demonstrated in the last few lectures, (b) review elementary Java operations including text input and output, loops and if/else statements, (c) make and call static methods, (d) use either several separate variables or one array to keep track of vote counts for different candidates. If you don't figure out how to use an array this way in the lab, this lab will make the problem much more clear. You will then learn it in the lecture easily.

Under the **`Lab10`** directory/folder make 6 subdirectories/folders named **`V1`**, **`V2`**, **`V3`**, **`V4`**, **`V5`**, and **`V6`** The preferred way to do this is the Unix command, given after going "in" **`Lab10`** with suitable **`cd`**'s:

```
mkdir V1 V2 V3 V4 V5 V6
```

However, a slower, less skillful way, is to (six times!) click the new folder icon and type V1, ..., V6 in DrJava when you begin work on each version. (Ask a neighbor or TA help you do this FAST.)

Start by saving the class definition file **`Election.java`** in the **`V1`** directory. Right after you finish AND TEST V1 (really easy!), do "save-as" GOING INTO the **`V2`** directory, to save **`Election.java`** to begin editing it to make it into version 2. Of course, do the same things for V3, V4, ..., V6!

For followup credit: When you are done with the lab work:

(1) Go to the directory/folder ABOVE **`Lab10`** (on Unix, command **`ls`** and see **`Lab10`**).

(2) Make a .zip archive of **`Lab10`** and all the subdirectories and files it contains. The Unix command for this is

```
zip -r Lab10.zip Lab10
```

(You should see the list of added files printed. IT'S WRONG if there are ANY ERROR MESSAGES!!) PC's, Macs, etc. have software to do this too. You probably know how to use it, and if not, you must find out if you want to complete the lab that way.

(3) For follow-up credit: Upload JUST THE **`Lab10.zip`** file We will give you full credit ONLY if your lab work history is captured in 6 separate version directories/folders (you may use more if you like.) EACH version subdirectory also must contain its version's **`Election.class`** file to prove you compiled each version after removing all syntax errors!

Instructions for TAs and for helping your classmates:

DO SHOW students how to make the directories and save versions, so they do it fast.

Do NOT show, write out, or dictate to the student ANY complete lines of code that go directly into the solutions to ANY of the steps.

But DO explain fully in English and give examples that are similar to what the step is intended to review or teach.

Page 190 of the text has a good example of if..else if..else if...else if...else.

Array technique idea: Make a length 6 array of ints with **`int a[]=new int[6];`**. Use the inputted vote number (0-4) AS AN INDEX to locate the array element you will use to count the candidate's votes, the votes for "other" or the invalid votes.

Finally, if your student's code shows the right idea but has a syntax error, DO EXPLAIN the syntax error and help the student correct it quickly!

ICSI201 Spring 2012 Lab 10
WHAT TO DO is printed here. HOW to do it is printed on the reverse.

BEFORE YOU BEGIN, read the reverse to SET UP 6 Version subdirectories/folders and understand lab expectations.

According to today's Wikipedia, the 2012 Republican primary candidates are Mitt (Romney), Rick (Santorum), Newt (Gingrich) and Ron (Paul). You will make a voting machine in Java today.

V1: Print **`Hello. Let's collect votes.`** Compile and TEST! (Points off if you DON'T have the resulting **`Election.class`** files in your .zip submission to prove you did the compiling!)

V2: Add to V1: Print **`The voting begins on the count of 10.`** Then write a for loop to make the computer print **`1`**, **`2`**, ... up to **`10`**, one number per line. Compile and test!!

V3: Add to V2: Make a **`static void`** method named **`printBallot`** that prints:
`Your choices: 0-Mitt, 1-Rick, 2-Newt, 3-Ron, 4-other,`
`5-To end the election now.`
And make your main method call it so you can test it. Compile and test!

V4: Add to V3: Make a **`static int`** method named **`getVote( )`**, and
(1) Code a call to **`getVote( )`** in **`main`** for testing purposes. Make **`main`** print out the return value for testing purposes. How? Simply code

`System.out.println(getVote( ) );`

(2) Make the **`getVote( )`** method call **`printBallot()`**
(3) Make the method then instantiate a **`Scanner`** with **`new Scanner(System.in)`** Of course, also declare a **`Scanner`** reference variable like **`sc`** so you can use your **`Scanner`**. Remember to

`import java.util.Scanner;`

and make sure you have **`sc =`** what it should so **`sc`** refers to the new **`Scanner`**.
(4) Make the method input the vote with **`sc.nextInt( )`**
(5) Make the method return the number the voter typed as the return value.

Compile, debug, and TEST!!

V5: After removing the call to **`getVote`** for testing purposes, add to V4 a loop in **`main( )`** to make it repeatedly get votes and stop the the first time somebody votes **`5`** Compile and TEST!

V6: Complete the V5 code. If you figure out how to use an array, do that because it is easier when you know how. If you don't, use 4 separate variables for the 4 candidates, one more for "other", and a 6th for invalid votes; and code a compound if-else statement like:
if(...) { ... } else if (...) { ... } else if (...) { ... } else if (...) { ... } else if (...) { ... } else { ... }
Make the program count how many votes each of the candidates 0, 1, 2, 3 got, count how many votes of 4 (for "other"), and count how many invalid votes (negative, or > 5).
When the voting is finished (somebody votes 5), how many of each kind of vote must be printed.

Compile, debug and TEST! Submitted **`.class`** file evidence of that will count!