

CSE21 : Lab #8 – Variables & Objects

Overview

This week will be just review and basic exercise to get you comfortable with OOP and variables. The common mistakes people tend to make are given in the sample and you will fix them. I suggest answering the questions at the same time as doing the lab since the questions are suppose to help you finish and understand the point of that particular exercise.

Getting started

You should have a Java project in Eclipse title Lab 21_7. This pdf is included in the project in the *doc* directory. The Java files you will use in this lab are in the *src* directory, as usual.

(Exercise) Fix – Lab21_Vars.java

Go through each question in this file and fix the error(s). 5.0 is a double but we want to assign it to an integer variable instead. We could of course change it to 5 but what if we couldn't? We can always type cast anything

Q1. How do you type cast a double into an int?

We want to declare an array of integers but instead have this in the code:

```
int arri0 = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};
```

Q2. How do you declare an array of int that goes from 10 to 1?

Variable are declared using <type> <name>. We want to share a variable in the if-clause and else-clause in the following code:

```
if (i < j) {  
    int temp = 0;  
    System.out.println("Temp is " + temp);  
} else {  
    temp = 1;  
    System.out.println("Temp is " + temp);  
}
```

Q3. What scope is the variable *temp* declared in? Where do you need it to be declared if it needs to be used for if-clause and else-clause?

Another common construct we use is a variable to store the total of some calculation. Suppose we want to do square of sums and want it in total which we later print out like the following code:

```
for (int i = 0; i < 10; i++) {
    int total = 0;
    total += i*i;
}
System.out.println("i value is " + i);
System.out.println("Total is " + total);
```

Q4. What scope does the variable *total* belongs in?

Q5. What gets print out as "i value is X"? And why is it that value?

We can create objects using *new* and also create pointers that references these objects. Multiple pointers can point to the same object or different objects. Analyze the following code to figure out the logical error.

```
Cheese jack;
Cheese monterey = new Cheese("Monterey");
jack = monterey;
System.out.println("Monterey name is " + monterey.getName());
jack.setName("Jack");
System.out.println("Jack name is " + jack.getName());
System.out.println("Monterey name is still " +
monterey.getName());
```

Q6. What is the logical error in the above code? (How do you fix it?)

Q7. How many pointers and objects are created in your fixed version?

Sometime we write code that duplicates or is redundant in different parts of our conditional statements. Here is one such example:

```
Scanner input = new Scanner(System.in);
System.out.print("Enter first number: ");

if (input.nextInt() > var3) {
    System.out.print("Enter second number: ");
    int num2 = input.nextInt();
    System.out.println("First is greater");
    if (num2 < var3)
        System.out.println("Second is Less than");
    else
        System.out.println("Second is Greater or equal");
} else {
    System.out.print("Enter second number: ");
    int num2 = input.nextInt();
    System.out.println("First is Less than or equal");
    if (num2 < var3)
        System.out.println("Second is Less than");
    else
        System.out.println("Second is Greater or equal");
}
```

}

Q8. What parts are redundant?

Q9. How do reduce it or combine it so we have no redundancy?

Q10. Suppose we want to print out the first number the user entered after all these lines. How can we figure out what the first number was? Add the `println` statement to print the number out.

(Exercise) Fill-in – Lab21_Objects.java

Take a look at `Dummy.java`. It has 10 constructors and 10 overloaded `display()` method. Your job is to call every one of them and fill in each of the `dlist` (Dummy list) entries with the appropriate objects. I have included the first two constructor calls and first two `display()` calls in the file. Put in 8 more constructor calls and 8 more `display()` calls.

DO NOT DECLARE NEW VARIABLES. YOU MUST USE ONLY THE VARIABLES GIVEN AS PARAMETERS TO THE METHOD CALLS. DO NOT USE ANY CONSTANTS OR NUMBERS AS PARAMETERS (ie 5, 5.0)

You can use individual indices of the arrays for the parameters. For example, `dlist[1].display(iarr[0]);` is good but `dlist[1].display(0);` is **not** as that is using a number directly as a parameter.

Q11. Give two distinct characteristics of a constructor:

Q12. What is the purpose of `\.` in `System.out.println` or `dlist[1].display();`?

Q13. What happens if you swap the order of the two lines? (and why?)

```
dlist[0].display(); // Goes first
dlist[0] = new Dummy(); // Goes after
```

What to hand in

When you are done with this lab assignment, you are ready to submit your work. Make sure you have done the following **before** you press Submit:

- ◆ Include answers to questions (Q1-Q12)
 - ◆ Attach fixed `Lab21_Vars.java`
 - ◆ Attach filled in `Lab21_Objects.java`
 - ◆ List of Collaborators
-