Lab 2

Name: ___________________  Checked: ______

Objectives:
• Learn about keyboard input to Java programs using the Scanner class.
• Practice using variables and assignment
• Experiment with simple arithmetic using the jGrasp Interactions pane
• Practice writing algorithms in pseudocode and considering alternative solutions

A. Implement a Java application  petName.java that uses variables to output a personalized message of the following form:

Hello, my name is Daphne and I am 18 years old. I’m enjoying my time at Villanova, though I miss my pet Luca very much!

Hints:
1) Include the following variable declarations, right at the beginning of the main method (i.e., before you print anything):

   String name = "Daphne";
   String petName = "Luca";
   int age = 18;

2) Incorporate the variables in your printing statements.

Check your work with a classmate – verify that petName.java works as stated.

Classmate signature: ____________________________

B. Now improve petName.java so that it obtains the values of the variables name, petName, and age as input from the keyboard.

Please enter name:
Anne Boleyn
Please enter pet name:
Purkoy
Please enter age:
32
Hello, my name is Anne Boleyn and I am 32 years old. I’m enjoying my time at Villanova, though I miss my pet Purkoy very much!

Check your work with a classmate – test each other’s programs to ensure they work well.

Classmate signature: ____________________________
C. Use the jGrasp Interactions pane to test some code snippets

Open jGrasp and click on Interactions tab (lower part of window).

You can type in expressions, for example (works like calculator):

- \( 4 + 3 \)
- \( 3.1 \times 0.2e-4 \)

... or Java statements such as variable declarations, assignment statements, and other simple Java code snippets.

- `int a = 1`
- `int b = 2`
- `int c = 3; // Note: semicolon is optional here`
- `a = c`
- `c = 5`

You can type any expression to get its value; type variable names to get their values:

- `a`
- `b`
- `c`

Try some more examples and note what you get:

- `14 / 3`
- `14 \% 3`
- `143 / 60`
- `143 \% 60`
- `8 / 12`
- `8 \% 12`
- `String word = "fish "`
- `String sentence = word + word`
- `sentence = sentence + sentence`
- `sentence = sentence + sentence` 
  (repeat this a few times and see what happens)

You can also experiment with Math:

- `Math.sqrt(2)`
- `double phi = Math.PI / 3`
- `Math.sin(phi)`

Notes about other things you tried: ______________________________________________

Check & discuss your work with a classmate.

Classmate signature: ______________________________________________

Tips:

- Watch the Workbench tab on the top/left part of the window; it lists your variables and their values.
- To avoid re-typing a line of code, use the up-arrow (one or more times)—it remembers the previous lines of code you entered.
- Java expressions that have a value can be evaluated directly. **Statements or directives that have no value need a semicolon.** Example:

  - `import java.util.Scanner;`
  - `if (a > 0) ans = "yes";`
D. We will create a Java application `Time1.java` that inputs values representing a time duration in hours, minutes, and seconds and then prints the equivalent total number of seconds. First, we construct an algorithm and then implement it in Java.

1) Algorithm:
Try to think of your own approach to doing this. Jot down your calculation for how you compute the answer for a particular instance of this problem, for example:

1 hour, 28 minutes, and 42 seconds is equivalent to _____________ seconds.

What were the steps that YOU used to calculate it? Now write these steps as an algorithm.

**Variables:**

**Algorithm:**

2) Java application
Implement your algorithm – start by typing it up as comments, then fill in actual Java code for each step in the algorithm.

*Check your work with a classmate*
- compare algorithms
- test each other’s programs at least twice (try some weird values)

**Classmate signature:** ________________________________
E. Create a Java application `Time2.java` that reverses the above process. That is, input a value representing a number of seconds, then print the equivalent amount of time as a combination of hours, minutes, and seconds. First, we construct an algorithm and then implement it in Java.

1) Algorithm:
Try to think of your own approach to doing this. Jot down your calculation for how you compute the answer for a particular instance of this problem, for example:

9999 seconds is equivalent to _____ hours, _____ minutes, and ____ seconds.

What were the steps that YOU used to calculate it? Now write these steps as an algorithm.

*Variables:*

*Algorithm:*

2) java application
Implement your algorithm – start by typing it up as comments, then fill in actual Java code for each step in the algorithm.

*Check your work with a classmate*
  * compare algorithms
  * test each other’s programs at least twice (try some weird values)*

*Classmate signature: ____________________________
Lab 2 Comments

Comments on this lab, please:

What was the most valuable thing you learned in this lab?

What did you like best about this lab?

Was there any particular problem?

Do you have any suggestions for improving this lab as an effective learning experience?