Lab 2

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

Preparation: Use variables to output a personalized message

1) Implement a Java program named PetName.java that simply prints the word “Hello.” Test it to ensure it works before proceeding.

2) Add some variable declarations, right at the beginning of the main method (i.e., before you print anything):

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

Substitute your name, age, and pet name. If you don’t have a pet, maybe try one of these: http://www.medievalists.net/2013/06/23/medieval-pet-names/

3) Modify your program to print the message below, incorporating the variables in your printing statements. Test your program to ensure it works correctly.

```
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!
```

4) Check your program for style and comments. If necessary, fix indentation or add some blank lines to make it more readable.

5) Submit PetName.java through blackboard under the assignment “Lab 2 Prep”
Part A: Input using Scanner

1) Run PetName and compare your work to your partner’s.

   Verify that the code works as stated, is well formatted and includes appropriate comments. If necessary, help your partner improve their code to make it more readable, then sign each other’s worksheet.

Classmate signature: _______________________________________

2) Now improve PetName.java so that it obtains the values of the variables name, petName, and age as input from the keyboard. Do this by inserting the appropriate code to use a Scanner to input value (Hint: refer to examples in lecture notes):

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

3) Experiment with your code some more and discuss with your classmate.

What happens if you enter a full name instead of just the first name? _________________
_____________________________________________________________________________
_____________________________________________________________________________

What happens if you enter the name instead of age or vice versa? _________________
_____________________________________________________________________________
_____________________________________________________________________________

Classmate signature: _______________________________________

Please enter name: Anne
Please enter pet name: Purkoy
Please enter age: 21
Hello, my name is Anne and I am 21 years old. I'm enjoying my time at Villanova, though I miss my pet Purkoy very much!
Part B: 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 * 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, sentence;
- word = "fish ";
- 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";
Part C: Writing algorithms

1) Write an algorithm to solve the following problem:

**Input values representing a time duration in hours, minutes, and seconds and then calculate and output the equivalent total number of seconds.**

[Hint: 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) Implement your algorithm as a java application **Time1.java**

Hint: Start by typing up the algorithm as comments for the program; 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: ________________________________

Villanova University     CSC 1051     www.csc.villanova.edu/~map/1051     Dr. Papalaskari
Part D: Writing algorithms

1) Write an algorithm to solve the following problem:

Input a value representing a number of seconds, then calculate and output the equivalent amount of time as a combination of hours, minutes, and seconds.

[Hint: This is the reverse of the previous algorithm. Try to think of your own approach for 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) Implement your algorithm as a java application Time2.java

Hint: Start by typing up the algorithm as comments for the program; 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

Name: ___________________________ Checked: _______

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?