Programming Fundamentals

Lab #1

Topics

- Basic Java syntax
- Eclipse IDE
- Storing and Manipulating Data
- Basic Input/Output

Concepts

println
class, main
comments
string concatenation
escape sequences
variable declarations, init, assignment
constants
primitive data types
arithmetic expressions (++,--,%,/,+,-,*,+=,-=,...)
data conversions (casting)
Scanner, System.in
nextLine,nextInt,nextDouble

Exercise 1

Use the "Eclipse IDE Tutorial" posted on BlackBoard to enter, compile, and run the following application:

```
public class Lab1Ex1 {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

After you are done, try to do the following and predict what will happen:
a) Change Lab1Ex1 to Hello
b) Remove the first quotation mark in the String
c) Remove the semicolon
d) Remove the last brace in the program

For each part, try to recompile and run your program. Note the error you are getting. Then fix the error by reversing what you did and rerun the program. Then try the next part.

Exercise 2

Make a Java program called GradeAvg that:
1. Creates 7 double variables named the following way:
quizScore1, quizScore2, quizScore3, testScore1, testScore2, quizScoreAvg, and testScoreAvg
2. Initializes those variables to % scores (make up some numbers between 0 and 100).
3. Initializes each to the corresponding average by using an appropriate expression.
4. Prints out the values of the variables containing the averages. Make sure to label each value.

Exercise 3

Create a new Java program called AgeGuess which
1. Creates a new Scanner object
2. Declares two variables:

```
String name;
int ageGuess;
```

3. Uses the Scanner object's methods to read in the name and ageGuess from the keyboard. Make sure to prompt the user for each value. The value entered should be on the same line as the prompt.
4. Displays the ageGuess and name variable values. Make sure to label each.

Exercise 4

Create a new Java program called TimeComp that reads a value representing a number of seconds, and then prints the equivalent amount of time as a combination of hours, minutes, and seconds. For example, 9999 seconds is equivalent to 2 hours, 46 minutes, and 39 seconds.