Programming Fundamentals

Lab #2

Topics

- Object instantiation
- Using existing classes (e.g. String, Random, Math, DecimalFormat)
- Pseudorandom number generation
- String methods, formatting
- Using a debugger
- Using conditional statements
- Constructing conditions and Boolean expressions

Concepts

new keyword
dot operator
method invocation
object reference variables
Strings – indexes, methods
Java packages
pseudorandom number generation, seed value
using static methods
formatting output
if, if-else, else statements
relational operators: ==, !=, <, >, etc.
Boolean operators: !, &&, ||
block statements
switch, break, default

Exercise 1

Modify the `AgeGuess` program from the last lab to do the following:

- Declare a new `int` variable `age`
- Initialize `age` to a random integer between 0 and 100 (inclusive)
- Asks the user for a guess, save the guess into the `ageGuess` variable
- Display the user guess and the correct answer

Exercise 2

Write an application called `DistCalc` that reads the (x, y) coordinates for two points then computes the distance between them using the following formula:

$$dist = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Display the result with three decimal places to the screen.

Exercise 3

Write an application called `UserNames` that reads the user's first and last name (separately), then prints a string composed of the first 4 letters of the user's last name, followed by the first letter of the user's first name, followed by a random number in the range of 10 to 99 (inclusive). You can assume the first name is at least one letter long and the last name is at least 4 letters.

Exercise 4

a) Modify the `AgeGuess` program from Ex. 1 by adding a conditional statement (if statement) to print out "You guessed wrong!" if the `age` and `ageGuess` variables are different. Remember that "not equal to" comparison is done using the NOT ($!=$) relational operator. Check to make sure the program runs without errors.

b) Add a nested if statement so that when the answer is wrong print out "older", if the age guess was less than the actual age, and "younger", otherwise. Check to make sure the program runs without errors.

c) Use the debugger (hit F11 in Eclipse) to run the program multiple times and check to make sure each of the different messages gets displayed. Remember to put a breakpoint first (CTRL+SHIFT+b).

Exercise 5

Make a Java program called `FloatEqu.java` and implement the following:

a) Declare a `double` variable and initialize it to `(1.0/10) * (1.0/10)`
Declare another `double` variable and initialize it to `(1.0/100)`
Insert an `if … else` statement and print out "EQUAL" if both variables are equal (use ==) and "NOT EQUAL" otherwise. Run the program and check the output. Is it what you would expect?

b) Modify the program by adding a conditional statement to determine if the variables are approximately equal, using the approach discussed in the lecture.

Exercise 6

Make a program called `NumDisplay.java` that prompts the user enter a number between 0 and 9 and then display the corresponding word (i.e. "zero" for 0, "one" for 1, etc.). Use a `switch` statement to do this. Include a default case that lets the user know they entered a wrong number.