Chad Philip Johnson
CSCI20, Worthington
August 26th, 2012

Pencil & Paper 03

1. Circle each of the following that is a legal variable name in Java: player2, playerTwo, 2player, player 2, PLAYERTWO, player2.java, p2, pLaYeR_tWo, player two, player.two
Legal Variable Names: player2, playerTWO, PLAYERTWO, p2, pLaYeR_tWo
Illegal Variable Names: 2player, player 2, player2.java, player two, player.two

2. Give the declaration of a variable called *number* of type *int*. Initialize the variable to the value *-1*.
The following code achieves this:

```
int number;
number = -1;
```

A more compact version does the same thing:

```
int number = -1;
```

3. What is the difference between a Java naming rule and a Java naming convention?
A Java naming rule requires that a variables, methods, classes, etc. use text/code in a particular way--deviating from one of these rules will produce a compilation error. A Java naming convention is a "recommendation" to produce better and more readable code, but will not produce a compilation error.

4. What are the three (3) Java naming conventions for variables?
Variables should: 1) begin with a letter and not an underscore "_" or dollar sign "$", 2) have names that are descriptive and self-documenting (e.g. "gasTank", "priceOfGas", "typeOfGas", etc.), 3) start with a lower case letter and have each proceeding word in the name starting with a capital letter (e.g. "amountOfMoneyToFillUpAGasTank", "gasCostsTooMuchMoney", etc.).

5. What is the output of the following line of Java code?

```
System.out.println( (int)3.14 );
```

Casting 3.14 to an integer causes all values after the decimal point to be dropped. The following line is printed:

```
3
```

6. In the previous question, what is the purpose of (int)? What is this operation called?
The code *(int)* is a *type cast* and allows values to be changed to different types. This allows greater control over assigning certain values to different types (e.g. a floating point value to an integer variable).

7. What is the output of the following line of code?

```
System.out.println( 21 / 5 );
```

The calculation returns an integer with any decimal values being discarded. The output is as follows:

```
4
```

8. What is the output of the following line of code?

```
        System.out.println( 21 % 5 );
```

The modulus operator returns the amount that remains following a division calculation. The output would be:

```
        1
```

9. Explain the difference between the operators / and %. Which of these operators works with variables of type int? Which of them works with variables of type double?

The division operator "/" works best with floating point values like *double*, where it returns a decimal value. When used with whole numbers values like *int* any decimal or remainder amounts are discarded and no rounding occurs. The modulus operator "%" works best with whole numbers values like *int*, where it returns the remaining amount from a whole number value division. It also works with types *float* and *double*, but this has less practical use.

10. What output is produced by the following statement?

```
        String message = "Lady XYZ";
        System.out.println( message.length() );
```

The output is the character length of the string:

```
        8
```

11. What output is produced by the following statement?

```
        String message = "Mister XYZ";
        System.out.println( message.charAt(7) );
```

The output is the seventh character (starting from zero) in the variable *message*:

```
        X
```

12. Write the statement to convert the String below to all lowercase letters.

```
        String greeting = "How Are You Today?";
```

The following command converts the string to have only lowercase letters:

```
        greeting.toLowerCase();
```

13. Write the statement to convert the String below to all uppercase letters.

```
        String sentence = "it is raining outside";
```

The following command converts the string to have only uppercase letters:

```
        sentence.toUpperCase();
```

14. What is the difference between the methods next and nextLine in the class Scanner?

The method *next* returns the next token according to the implemented delimiter (default is one whitespace character " ") whereas the method *nextLine* proceeds through the input to the next line separator and returns all skipped tokens.

15. What is the output of the following statement?

```
        double average = 97.89521;
        System.out.printf("The average is %5.2f", average);
```

The output is as follows:

```
        The average is 97.90
```

16. Write an output statement to display the variable exactPrice below in a field of width 8, rounded to 4 decimal places.

double exactPrice = 22.1257999;

The code would be as follows:

System.out.printf("%8.4f", exactPrice);

17. What is a comment in Java source code?

A comment is a line of text within source code that the programmer uses to make notes about the code and to provide details about how the program works. Comment lines are ignored by the compiler and have no effect on actual code.

18. Explain the difference between these two commenting styles in Java -- /* */ and //.

The comment form /* */ allows for as much text and line breaks to occur as necessary in between the two asterisks. The double forward slash // comment form allows text to be written until a line break occurs which ends the comment.