COP 2250 – Java Programming 1 – Chapter 4 – Math, Characters, & Strings

The Math Class

* This class is in the java.lang package and so is always available in your programs.
* There are many methods and a few constants. See pages 120-124.
* NOTE that the trig methods measure angles in radians, not degrees.
* The Math.random() method is an important method for generating random numbers.
* With casting to (int), random integers can be created. See page 122.

The Random Class

* This class in the java.util package can also generate random numbers.
* It has methods for creating random numbers in various numeric data types. See page 335.
* This class is tested on the 1Z0-811 Exam.

Try ComputeAngles

The Character Data Type

* This data type stores a single character, using the **char** keyword.
* Use single quotes (apostrophes) when assigning a char.

char ch = ‘A’;

* Characters can be encoded using ASCII or Unicode.
* ASCII is an 8-bit scheme with 256 possible characters. See Table 4.4 and Appendix B.
* Unicode is a 16-bit scheme so 65,536 characters were possible.
* Even that could not handle all characters in all languages so Unicode was extended.
* You can use either system:

char ch = ‘a’; // ASCII way

char ch = ‘\u0061’; // Unicode equivalent (recall that hex 61 = 97)

* char is actually a numeric type. See Note on page 126.

Escape Sequences

* See page 126 and Table 4.5

Casting between char and Numeric Types

* This is possible because char is a numeric type.
* See page 127 for examples.
* Implicit casting without ( ) works if the cast fits into the target type, otherwise explicit casting with ( ) is required.
* Numeric operators can be applied to char variables.

Testing Characters

* You could use if expressions with the ASCII range for the sought characters as shown just above Table 4.6 on page 128.
* However, Table 4.6 shows the Character class methods that make this task much easier.

The String Type

* String is a defined class in the Java API.
* String variables are reference, (object) types.
* The value stored in a string variable is the address of the string in memory.
* In contrast, numeric types like int, double etc. are primitive types. The data they store is the assigned numeric value.
* Table 4.7 shows some simple methods of the String class. There are many more in the API.
* Study well the String skills and examples on pages 130-138. You will need these skills often.
* NOTE well the difference between using the **next()** and **nextLine()** methods of Scanner for inputting strings. The Caution in the middle of page 133 covers the danger.
* Table 4.8 shows the String class comparison methods.
* Note well the Caution for comparing strings with relational operators on page 134. These operators would compare the addresses of the strings, not the strings themselves.

Try OrderTwoCities

* Tables 4.9 and 4.10 show the String class methods used for finding substrings.

Try GuessBirthday

Try HexDigit2Dec

Try LotteryUsingStrings

Formatting Console Output

* Control of decimal places in numeric output can be accomplished with System.out.**printf()**.
* General format is:

System.out.printf(“formatting string”, item(s) to format);

* The formatting string contains **format specifiers**, one for each item to be formatted.
* Table 4.1 on page 146 displays the commonly used format specifiers.
* The items must match the type and sequence of the format specifiers in the string.
* Note well the example at the very bottom of page 146.
* Table 4.12 on page 147 displays specifiers for width and precision.
* Note the important Caution and Tip on page 148.

Try FormatDemo