Lesson 8 (Character, String and Number Methods)

1. Java Number class is an example of a wrapper class which converts a primitive data type to an object of the class. Note that the number class is not a real class but an abstract class.

2. Methods in Java Number Class

   (a). compareTo function compares a given object to a different object of the same type. If both the values are the same then the function returns 0. If the given number is less than the argument then it returns -1. Else it returns 1.
   (b). equals() method checks if the number and the argument passed through the method is not null and equal to the number. If both the numbers are equal, it returns true. If not, it returns false. They also have to be of the same type.
   (c). Java toString method is used for converting a numeric datatype to a string. This helps in returning the value of an object. Calling this method means, overriding the. toString() method.
   (d). valueOf , this method returns the number object which is relevant. Simply put, this method converts the argument passed through it to an Integer object.

   There are three alternative ways to use this method:
   - valueOf(int i)- This returns an Integer object which represents the integer value.
   - valueOf(String i)- This returns an Integer representation of the string passed
   - valueOf(String i,radix)-This returns an Integer representation of the string passed of base radix.

Method	Description
`isUpperCase()`	Tests if character is uppercase
`toUpperCase()`	Returns the uppercase equivalent of the argument
`isLowerCase()`	Tests if character is lowercase
`toLowerCase()`	Returns the lowercase equivalent of the argument
`isDigit()`	Returns `true` if the argument is a digit (0–9) and `false` otherwise
`isLetter()`	Returns `true` if the argument is a letter and `false` otherwise
`isLetterOrDigit()`	Returns `true` if the argument is a letter or digit and `false` otherwise
`isWhitespace()`	Returns `true` if the argument is whitespace and `false` otherwise; this includes the space, tab, newline, carriage return, and form feed

A. The toUpperCase () method and toLowerCase () method convert any String to its uppercase or lowercase equivalent, respectively.
B. The length() method returns the length of a String. For example, the following statements result in the variable len that holds the value 5.
    i. String greeting = "Hello";
    ii. int len = greeting.length();
C. The indexOf() method determines whether a specific character occurs within a String. If it does, the method returns the position of the character; the first position of a String begins with zero. The return value is –1 if the character does not exist in the String.
D. The endsWith() method and the startsWith()method each take a String argument and return true or false if a String object does or does not end or start with the specified argument.
E. The class java.lang.Math contains constants and methods that you can use to perform common mathematical functions. All of the constants and methods in the Math class are static—they are class variables and class methods.
F. Math.max() method returns the larger of two values, and the method Math.abs() returns the absolute value of a number. The statement largerValue = Math.max(32, 75); results in largerValue assuming the value 75, and the statement posVal =Math.abs(-245); results in posVal assuming the value 245.

Method	Value That the Method Returns
`abs(x)`	Absolute value of *x*
`acos(x)`	Arc cosine of *x*
`asin(x)`	Arc sine of *x*
`atan(x)`	Arc tangent of *x*
`atan2(x, y)`	Theta component of the polar coordinate (r, theta) that corresponds to the Cartesian coordinate x, y
`ceil(x)`	Smallest integral value not less than *x* (ceiling)
`cos(x)`	Cosine of *x*
`exp(x)`	Exponent, where *x* is the base of the natural logarithms
`floor(x)`	Largest integral value not greater than *x*
`log(x)`	Natural logarithm of *x*
`max(x, y)`	Larger of *x* and *y*
`min(x, y)`	Smaller of *x* and *y*
`pow(x, y)`	*x* raised to the *y* power
`random()`	Random `double` number between 0.0 and 1.0
`rint(x)`	Closest integer to *x* (*x* is a `double`, and the return value is expressed as a `double`)
`round(x)`	Closest integer to *x* (where *x* is a `float` or `double`, and the return value is an `int` or `long`)
`sin(x)`	Sine of *x*
`sqrt(x)`	Square root of *x*
`tan(x)`	Tangent of *x*