Java Programming for Beginners

Lab Exercise 8

Answer Key

1) Java provides class implementations of primitive objects –**int** and **Integer**, for example—but the String class has no primitive counterpart. Why not?

> Strings are not appropriate for a primitive implementation because they are of variables sizes. A defining trait of primitives is that they take up a fixed amount of space in memory. If String was a Java primitive, it would need to have a maximum length.

2) The Object class describes a *hashCode()* method. What is this method's intended purpose? How does it relate to the *equals()* method?

> The *hashCode* method exists to convert objects into integer values which are unique to the values of the object. In a perfect Java world any two objects of the same type with are "equal" according to the *equals()* method will also generate the same hashCode.

3) We can create our own Java exceptions. Write an exception class of your own by inheriting from the **Exception** class. Now, create another class that inherits from the **RuntimeException** class. What is the difference between these two classes?

> The **Exception** class should be inherited from if we would like to create a **Checked Exception**. That is, an exception that must be explicitly handled by our code if it might be thrown. **RuntimeException**, however, is an **Unchecked Exception**. Exceptions of this type do **not** have to be handled when they might be thrown (though, of course, they can be!)