Virtual Lecture Notes (Part 1)

Overview of the Scanner Class

The **`Scanner`** class includes methods for accepting user input from the keyboard as indicated in the following abbreviated Method Summary table.

double	**`nextDouble`**() Scans the next token of the input as a `double`.
int	**`nextInt`**() Scans the next token of the input as an `int`.
String	**`next`**() Finds and returns the next complete token from this scanner.
String	**`nextLine`**() Advances this scanner past the current line and returns the input that was skipped.

Notice that the Java API format of the **`Scanner`** class and the **`String`** class follows the same basic pattern and includes the following information.

1. The name of the method.
2. A description of the method.
3. A parameter list, which in this case is empty for all four methods.
4. The return type of the method (on the left).

These four methods, **`nextInt()`**, **`nextDouble(),next()`**, and **`nextLine()`** can accept simple numeric or alphanumeric input from the keyboard. Which one of the **`Scanner`** class methods would you use to accept a decimal value entered by a user?

Before these methods can be used, a **Scanner** object must be created. Once a **Scanner** object exists, using **Scanner** class methods will follow the same pattern as calling **`String`** class methods (e.g. **`object.method()`**).

Because Strings are used so frequently, the Java developers decided to use a shortcut for declaring them that is similar to the way **`ints`** and **`doubles`** are declared.

```
int testScore = 96;
double interestRate = 0.045;
String lastName = "Swarzenegger";
```

However, this shortcut only works with **`String`** objects; an intermediate step is required to declare objects of most other classes. For example, a **`Scanner`** class object can be created as follows.

```
Scanner keyboardInput = new Scanner(System.in);
```

This statement uses Java's reserved word **new** to construct a **Scanner** object named **keyboardInput**. With a Scanner object declared, the following statement could be written to accept a decimal number typed in from the keyboard.

```
double number = keyboardInput.nextDouble();
```

Does this statement look familiar? Dot notation is used to separate the object on the left from the method on the right. When this statement is executed, the program invokes the **nextDouble()** method on the **keyboardInput** object by pausing to wait for the user to enter a decimal number and then press the Enter key. Once the input is typed and the Enter key is pressed, the decimal number will be assigned to the identifier called **number**. Since **keyboardInput** is an object of the **Scanner** class, it can invoke any of the methods of the **Scanner** class.

There are a lot of details we are ignoring for now about objects, but that doesn't mean you can't begin using the methods of the **Scanner** class to accept user input.