The String Class

- No import needed
- Straddles the worlds of primitives and objects
  - String myword = “Bob”; //This is called a String Literal
  - String myotherword = new String();
- Always use the .equals method or the .compareTo method instead of == when comparing two strings
  - if (myword.equals(myotherword))
The process of “adding” strings together

```java
String myname = "Bob is ";
String myage = " years old."
int x = 10;
String answer = new String();
answer = myname + x + myage;

//note the int is converted to a String
```

The plus sign is the only overloaded operator in Java
Useful Methods from the String Class

- `int length()`
  - `int x = myword.length();`

- `boolean equalsIgnoreCase(String string)`
  - `if (myword.equalsIgnoreCase(theirword))`

- `String replace(char old, char new)`
  - `myword = myword.replace('a', 'A');`

- `String toLowerCase()`
  - `myword = myword.toLowerCase();`

- `String toUpperCase()`
  - `myword = myword.toUpperCase();`

- `String trim()`
  - `myword = myword.trim();`
Methods to Find Characters

- int indexOf(char character)
  - int x = myword.indexOf(‘a’);
- int indexOf(String string)
  - int x = myword.indexOf(“ing”);
- int indexOf(char character, int beginIndex)
  - int x = myword.indexOf(‘x’, 3);
- int indexOf(String string, int beginIndex)
  - int x = myword.indexOf(“ax”, 5);
- char charAt(int index)
  - char myletter = myword.charAt(3);
Methods to Get Part of a String

- `String substring(int beginIndex)`
  - `String myword = myphrase.substring(3);`

- `String substring(int beginIndex, int endIndex)`
  - `String myword = myphrase.substring(2, 8);`
  - //note that this returns the substring ending at position `endIndex - 1`
The Scanner Class

- import java.util.Scanner;
- Creating an instance of the Scanner class...
  - Scanner myin = new Scanner (System.in);
    //prepares to read from the keyboard
  - Scanner myfile = new Scanner (new File("sample.txt"));
    //prepares to read from a text file
  - Scanner words = new Scanner (mysentence);
    //mysentence is a String object, and we’ll be able to use this to break it in to words
Scanner Methods

- int `nextInt()`
- double `nextDouble()`
- String `nextLine()`
  
  //since `nextInt()` and `nextDouble()` do not “consume” the new line character that follows them, and extra `nextLine()` is required if you are reading text after reading numbers
- String `next()`
- boolean `hasNext()`
- boolean `hasNextInt()`
- And many more...
  - [Scanner JavaDoc](Scanner JavaDoc)
import java.util.Scanner;

public class Prob1
{
    public static void main(String[] args)
    {
        Scanner myin = new Scanner(System.in);
        int mynum;

        System.out.println("Please enter a number: ");
        mynum = myin.nextInt();
        System.out.println("The number you typed was 
" + mynum);
    }
}
import java.util.Scanner;

public class Prob1
{
    public static void main(String[] args)
    {
        Scanner myin = new Scanner (System.in);
        String myword = new String( );

        System.out.println("Please enter a word:");
        myword = myin.next();
        System.out.println("The word you typed was " + myword);
    }
}
import java.util.Scanner;
import java.io.*;

public class Sample {
    public static void main(String[] args) {
        try {
            Scanner myfile = new Scanner(new File("sample.txt"));
            int mynum;

            while (myfile.hasNextInt()) {
                mynum = myfile.nextInt();
                System.out.println(mynum);
            }
            myfile.close();
        } catch (IOException e) {
            System.out.println("File Error: " + e);
        }
    }
}
Your To Do List

- Read 68 – 70, 264 – 269
- Review Book Section 2.10
- CodingBats – String1
  - withoutEnd
  - extraFront
  - frontAgain
- Lab – WordCount (2 parts)
The WordCount Lab

- **Part I – Console Application**
  - Prompt the user to enter a phrase with no punctuation
  - Calculate and display the total number of characters
  - Calculate and display the average length of the words

- **Part II – Input from File**
  - Create a text file that contains a phrase with no punctuation
  - Modify your program from Part I so that the phrase is read from the file instead of the keyboard

- **There will be a Part III and Part IV as well**