CS 61B
Jonathan Shewchuk  jrs@cory.eecs
Prof & all TAs at once (preferred): cs61b@cory.eecs

Reading: Sierra & Bates pp. 1-9, 18-19, 84.
http://www.cs.berkeley.edu/~jrs/61b
Piazza board is required reading.
Labs start tomorrow. Attend your scheduled lab.

Grading

<table>
<thead>
<tr>
<th>Points</th>
<th>Component</th>
<th>Grade Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Labs</td>
<td>185+ → A+</td>
</tr>
<tr>
<td>20</td>
<td>Homeworks</td>
<td>175-185 → A</td>
</tr>
<tr>
<td>70</td>
<td>Projects</td>
<td>165-175 → A-</td>
</tr>
<tr>
<td>25</td>
<td>Midterm I</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Midterm II</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>Final Exam</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No Code Rule

1-line exception: Compiler error msg
→ cut & paste entire error msg including line of code.
Goals of CS 61B

1) Learning efficient data structures & algorithms.
2) Designing & writing large programs.
3) Understanding & designing data abstraction & interfaces.
4) Learning Java.
Object-oriented programming

Object: A repository of data.
Class: Type of object.
Method: Procedure or function that operates on an object or class.

addItem: adds an item to any ShoppingList object.

Inheritance: A class may inherit properties from a more general class. ShoppingList inherits from List the property of storing a sequence of items.

Polymorphism: One method works on several classes, even if the classes need different implementations.

e.g. "addItem" method on every kind of List, though adding item to a ShoppingList is different from a ShoppingCart.

Object-Oriented: Each object knows its own class & methods. Each ShoppingList & ShoppingCart knows which implementation applies to it.

Java

Variables: you must declare them and their type.

Python: `x = 1`  Scheme: `(let ((x 1)) )`

Java: `int x;`  
\[ x = 1; \]

Does 2 things:
1. Allocates memory to store an integer, type "int".
Names variable "x". Variables also used to reference objects.

2 ways to get classes:
1. Use one defined by somebody else. Java has tons.
2. Define your own.

```java
String myString; // class built into Java.
Can store a reference to a String object.

myString = new String();
```

2 steps:
- "new String()" is a constructor.
- assignment "=" causes myString to reference the object.

```
myString = new String();
```
Java programs must be compiled before you can run them.

**Python**

Python program (.py)

```python
python
Answer
```

**Java**

Java program (.java)

```java
javac
.class files
java [Java Virtual Machine (JVM)]
Answer
```
Objects & Constructors

```java
String s;
// Step 1: declare a String variable
s = new String();
// Steps 2, 3: construct empty String; assign it to s.

s = "YoW!");
// Steps 1, 2, 3 combined.

String s2 = s;
// Now s & s2 reference same object.

s2 = new String(s);
// Now referencing 2 different, identical objects

- Look where s points.
- Follows reference to string object.
- Reads String
- Constructs new String w/copy of characters
- Makes s2 reference new String.
```

3 String constructors:
1. new String() constructs an empty string (zero characters)
2. "YoW!")
3. new String(s) takes a parameter s. Makes copy of object that s references.

 Constructors always have same name as their class, except "stuffinquotes".
```java
METHODS
s2 = s.toUpperCase();

s2 \[\rightarrow\] YOW!

String s3 = s2.concat("!!"); // s3 = s2 + "!!";

s3 \[\rightarrow\] YOW!!

String s4 = "*".concat(s2).concat("*"); // s4 = "*" + s2 + "*

s4 \[\rightarrow\] **YOW!**

The object "Yow!" did not change, unchanged.

Strings are immutable. Their contents never change.

s2 = s2 + s2;

YOW! YOW!
I/O classes & Objects

Objects in System class for interacting with a user:

- System.out is a PrintStream object that outputs to the screen.
- System.in is an InputStream object that reads from the keyboard.

readLine is defined on BufferedReader objects.

- How do we construct a BufferedReader? With an InputStreamReader.
- An InputStreamReader? With an InputStream.
- An InputStream? System.in is one.

Figure this out via online Java libraries API — java.io

\( \rightarrow \) Reads raw data.
\( \rightarrow \) Compose into characters (2 bytes long)
\( \rightarrow \) Compose into entire lines of text.

```java
import java.io.*;

public class SimpleIO {
    public static void main(String[] args) throws Exception {
        BufferedReader keyboard = new BufferedReader(new InputStreamReader(System.in));
        System.out.println(keyboard.readLine());
    }
}
```
To use Java libraries, other than `java.lang`, you “import” them.

`java.io` includes `InputStreamReader`, `BufferedReader`, etc.

Java program always begins at a method called “main.”
import java.net.*;
import java.io.*;

class WHWWW {
    public static void main(String[] arg) throws Exception {
        URL u = new URL("http://www.whitehouse.gov/");
        InputStream ins = u.openStream();
        InputStreamReader isr = new InputStreamReader(ins);
        BufferedReader whiteHouse = new BufferedReader(isr);
        System.out.println(whiteHouse.readLine());
    }
}
