You're a Programmer?!

Yea, I know Java Pretty well...

Wow! Can you make me a game?!

One does not simply MAKE a game...
Announcements

- HW8 will be graded today

- HW10 is due next Monday (not Friday)
Highlights

- Java

```java
public static void main(String[] args) {
    for(int i=0; i < 20; i++)
    {
        System.out.println("i is: " + i);
    }
}
```
Similarities

Java is similar to C++ and should not be too hard to go between these languages.

Java (and most programming languages) have the basic tools:
1. if/else
2. arrays
3. functions
4. classes (if object oriented)
Differences

Easier:
  Automatic memory management
  No pointers
  No pass-by-reference (the & in functions)
  No operator overloading
  Automatically initializes variables

Harder:
  Stricter types
  Stricter structure of programs
Differences

Java has automatic memory management, so there is no delete

However, “new” still exists and you use it the same place as C++: when you want boxes

Java also does not technically compile, instead it run code on a virtual machine (this makes it more portable, but also slower)
Differences

The way Java and C++ represent data is slightly different:

In C++:
  Pointers & Arrays = memory address
  Types = boxes

In Java:
  Classes & arrays = memory address
  int/double/char/etc. = boxes
Hello world!

Let's start with the basics, printing to terminal

This might seem weird, but in Java everything must be in a class (EVERYTHING)

So to do simple printing, we must first make a class, put main inside it (in a special way), then we can finally print:

```
System.out.println("Hello world!");
```

(see: helloWorld.java)
Types

Types are much more strict in Java, and you cannot convert between them as easily.

It still allows int -> double conversion, but you cannot go double -> int without type casting.

Conversions between char and int are still fine (though less useful in Java).

(see: strictTypes.java)
Static

Other than forcing things to live inside a class, Java works in a very similar manner to C++

Coming from C++, the concept of “static” might seem a bit odd

This is a property of the class, shared by all objects/instances of the class (thus there is only “one” of these shared by everyone of this type)
Consider this:

```java
public class Example {
    public static int shared;
    public int mine;

    Example x = new Example();
    Example y = new Example();

    x.mine = 2;
    x.shared = 3;

    y.mine = 20;
    y.shared = 30;
}
```

(see: Example.java)
Standard tools

Other than this, Java has the same normal tools C++ does.

Array notation is slightly different, where the array is denoted on the type and not name: (and they are always dynamic)

```java
int[] x = new int[20];
```

(see: Tools.java)
Inheritance works pretty much the same way as we talked about it in C++ (though there are subtle differences of more advanced stuff).

Java by default has every function virtual, along with always being a memory address makes inheritance use a lot easier.

(see: ClassesAndInheritance.java)
Mini-problem

For the rest of the time, I will solve this midterm 2 sample problem in Java:

**Problem (5) [15 points]** Write a C++ code segment that reads a sentence from the keyboard. You then need to write a modification of this sentence into a file called ”garble.txt”. The modified sentence you write to the file should do two things: (1) change any lower case ”e” to two upper case ones and (2) add ”z” to the end of every word (i.e. before a space). An example sentence and what should be written to the file is provided below.

<table>
<thead>
<tr>
<th>Sentence</th>
<th>File result</th>
</tr>
</thead>
<tbody>
<tr>
<td>hello thEre, human</td>
<td>hEEElloz thErEE,z humanz</td>
</tr>
</tbody>
</table>

(see: StringMash.java)