My Richard The Third The Video Game is going to be FANTASTIC.

MAN I HEAR THAT

All I need to do is program it!

LATER:

Um, APPARENTLY, programming is for folks who are thrilled when a computer reminds them they're missing a bracket or semicolon? It must be, because they make that happen SO OFTEN.

So it's not going well?

I CAN'T EVEN GET RICHARD THE THIRD TO MOVE. You know what my game is now? My game is NIBBLES, with the text changed from "Copyright Microsoft 1990" to "man, forget this"

Programming's a skill!

I just thought it was a skill I could pick up easily. I don't need to know everything! I don't need to know the difference between friggin' binary and B+ search trees! ALL I WANT TO KNOW is how to make Richard III's sucky horse do double jumps, you know? You've got to learn to crawl before you can run, T-Rex.

Hey, here's a tip!

Crawling sucks!
Java Basics

- Input/Output
- Data Type Conversions
Data Output

- Use: System.out.println()

- API documentation for 'System' is here: http://docs.oracle.com/javase/7/docs/api/java/lang/System.html

- API documentation for 'println()' is here: http://docs.oracle.com/javase/7/docs/api/java/io/PrintStream.html
Data Output

• For numbers with strings:

    String year = “2014”
    System.out.print(“The year is: ” + year)
Data Input

• For strings:

    Scanner reader = new Scanner(System.in);
    System.out.println("Enter input: ");
    String input=reader.nextLine();

• For numbers:

    Scanner reader = new Scanner(System.in);
    System.out.println("Enter an integer");
    int input=reader.nextInt();
Data Output

- Mixing numbers with strings:

```java
String year = "2014";

// Not what we want (string concatenation):
System.out.println("Next year is: " + (1 + year);

// Convert string to int then add:
int nextYear = 1 + Integer.parseInt(year);
System.out.println("Next year is: " + nextYear);
```
Type Conversion

- In certain circumstances your data types may need to be converted to other data types (e.g. converting a string into an integer or vice versa).

- **String → int**
  
  ```java
  foo = "31337";
  int bar = Integer.parseInt(foo);
  ```

- **int → String**
  
  ```java
  Integer.toString(bar)
  ```
Type Conversion (more)

- Variables are declared as a specific type
- Variables are that type for the life of that variable
- Variable types cannot be changed once declared

String year = “2014”;
// errors (because year is forever a String):
year = Integer.parseInt(year);

// instead, create an int variable to store it in:
int iYear = Integer.parseInt(year);
Writing Programs (Eclipse)

- New Java Project
- New Class
- Save
- Run as java program
API
(Application Programming Interface)

- http://docs.oracle.com/javase/6/docs/api/
- System.out.println():
  Java.lang → System → out → println()
Packages/Modules/Libraries

- Libraries store code for reuse later.
- Imported for use when needed.
- Why modules:
  - Black box coding (limited scope of knowledge)
  - Keep programs reasonably sized for editing
  - Code-reuse (DRY – Don't Repeat Yourself)
// Get access to swing (GUI) functions
import javax.swing.*;

/**
 * Example of importing to gain extra functionality
 * @author acase@cs.nyu.edu
 */
class ImportExample {
    public static void main(String[] args) {
        OptionPane.showMessageDialog(null, "Hello again!");
    }
}