Staff

Hau Lian
Walter Menendez
Joey Putnam
Ada Taylor

6.s092@mit.edu (goes to all four of us)
Class Details

Piazza: http://piazza.com/mit/spring2014/6s092

● 9 lectures, 4 assignments
● You have to write your own code!
● **PASSING:** complete and pass four assignments
● Bring your laptops!
What we’ll do

Learn Java

Learn how to use Eclipse

Learn some programming concepts that will appear in 6.005
What we’re assuming

You know how to program
You’re taking 6.005 eventually
Resources

- 6.005 Elements of Software Construction site: http://stellar.mit.edu/S/course/6/fa13/6.005/
- Sun Java Tutorial http://docs.oracle.com/javase/tutorial/index.html
- Effective Java, Bloch.
- Java in a Nutshell, 5th Edition, by Flanagen
- Safari http://libraries.mit.edu/get/safari
- Books 24x7 http://libraries.mit.edu/get/books24x7

Google is your friend!
What is Java?

1991 - 1995 Originally developed by James Gosling at Sun Microsystems (later merged into Oracle 2009 - 2010)

- Aimed to have a familiar C/C++ style notation and architecture neutrality, “Write Once, Run Anywhere”
- Became popular with the ability to run Java applets within web pages
- 2006 - 2007 Sun released Java as free and open source software (FOSS)

Used by over 9 million developers!
Let’s visit Eclipse!
Java is compiled

You write source code that you save with a .java extension.

You compile that into a .class file to run your program.
Hello World!

class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello world!");
    }
}


Program template

class CLASSNAME {
    public static void main (String[] args) {
        STATEMENTS
    }
}
}
Output and comments

// This is a comment

/* This is also a comment */

System.out.println("And this is getting printed to console"); // NOTE THE SEMICOLON
Java Roadmap

Data
Data structures
Operations
Control flow
Types

int: Integer (1, 0, 42, -1248)
double: Real number (3.14, 1.0, -0.8)
String: Text consisting of characters ("hello", "MIT")
boolean: Truth value (true or false) // note the lowercase
Variables

General syntax: `type` name (= value);

String name;
ArrayList<RandomObject> list;
Conventions for naming

- Case sensitive
- An unlimited-length sequence of Unicode letters and digits, beginning with a letter, the dollar sign "$", or the underscore character "_"
- White space is not allowed
- Cannot use any of the 50 reserved words or keywords (e.g. class, int, void)
Assignment

Java is “statically-typed” so all variables must be declared before being used (otherwise you’ll throw an exception!)

Insufficient in Java:
lastName = “Menendez”;

Instead:
String lastName = “Arjuna”;
= is for assignment, as in the mathematical form ‘Let int myInt = 4’
Re-assignment

This is also valid:

```java
int todayLow = 6;
todayLow = 12;
```
Conversion

double score = 3.5;
int otherScore = score + 5; // INVALID

int simpleScore = 2;
double copy = 2; // This is fine! No data is lost.
Casting

Sometimes you have to force it.
double score = 3.5;
int simpleScore = (int) score; // simpleScore = 3

...you can’t force everything.
int num = (int) “String!”;
Binary Operators

int x = 4;
out.println(x++); //postfix, outputs 4
out.println(++x); //prefix, outputs 6!

“Do a thing and then increment” vs. “Increment and then do a thing”
public static void main(String[] args) {
    String text = "Lucky" + " numero ";
    text = text + 2 + "!";
    out.println(text);
}
Order of Operations

Assignment: =
Addition: +
Subtraction: -
Multiplication: *
Division: /
Modulo (integers only!): %
Conditionals

== Equal
!= Not equal
> Greater than
>= Greater than or equal to
< Less than
<= Less than or equal to
Control flow

What we’d like:
1. Do something only when X is true
2. Do something a certain number of times
3. Keep going or come back to a line of code

1 + 2 today, 3 next time
Decision-making

if (Boolean expression) {
    STATEMENTS
}

if (isValid)
    if (numFriends > 4)
Fancy: ternary conditional

For dynamic assignment

Type var = expression? valIfTrue : valIfFalse
int x = 5;
int y = x <= 5 ? 6: 7;
System.out.println(y); // Will print 6
Decision making, cont.

if( ...) {
    // ...
} else if (...){
    // ...
} else {
    // ...
} else {
    // ...
}
Decision making example

```java
if (test < 0){
    out.println("You might want to throw an error..."ожно);
} else if (test > 2) {
    out.println("Test seems too big!");
} else {
    System.out.println("Just right!");
}
```
Loops #1

while loops
while( this is true) {
    // do things
}

while (test < 5){
    out.println("Do a thing!");
    test++;
}
Loops #2

for loops gather the initialization term, termination condition and increment operation into one place

for (initialization; termination; increment) {
    statements
}

\[
\text{for(int i = 0; i < 3; i++)}
\]

\[
\text{out.println("test and i = " + test + " " + i);}
\]
A quick word on scope

```java
boolean myBool = true;
while (myBool) {
    out.print("This will print once");
    myBool = false;
    String localVar = "This exists only in here!";
}

out.println(localVar);```

Arrays

A primitive data structure!
Type[] arrayName = new Type[INT_SIZE];

Arrays are of a fixed size. (Coming soon: ArrayList)

Zero-indexing
Arrays, cont.

int[] nums = new int[7]; // using an int
int x = 5;
int[] nums = new int[x]; // using a variable
int[] nums = new int[4*3 -2]; // using an expression

Use any way of declaration!
Arrays, cont.

Go ahead, initialize.

Type[] array = {/* LIST OF VALUES */};

From Assgn. 1 Test:

String[] test = {"JAVA", "MIT"};
Assignment 1 is due Monday!