

6.092: Introduction to Java

# 1: Types, Variables, Operators

# Class Details

Stellar web site:

<http://stellar.mit.edu/S/course/6/ia11/6.092/>

6 Lectures + Assignments

# Goal

Learn enough Java to do something useful

Examples:

- Simulate a natural/engineering process
- Manipulate data in some way
- Simple web applications

# Assignments

- View and submit via Stellar
- Due at 3 PM the next day (24 hours)
- Collaborate with others
- Write your **own** code
- Must submit first assignment

Must submit a “reasonable” attempt for 5/6 assignments to pass

# CPU Instructions

$z = x + y$

Read location x

Read location y

Add

Write to location z

# Programming Languages

- Easier to understand than CPU instructions
- Needs to be translated for the computer
- Programs describe a process (“how to do x”) very precisely

# Java

- “Most popular” language
  - More complex than some (eg. Python)
  - Simpler than others (eg. C++)
- 
- Server apps (Gmail)
  - Mobile apps (Android)
  - Business apps (SAP)

# First Program

```
class Hello {  
    public static void main(String[] arguments) {  
        // Program execution begins here  
        System.out.println("Hello world.");  
    }  
}
```



# Program Structure

```
class CLASSNAME {  
    public static void main(String[] arguments) {  
        STATEMENTS  
    }  
}
```

# Output

`System.out.println(“your text here”)`  
outputs to the console

Example:

```
System.out.println("output");
```

# Comments

// this text is ignored

Comments are notes for humans  
Both yourself, and others

# Second Program

```
class Hello2 {  
    public static void main(String[] arguments) {  
        System.out.println("Hello world."); // Print once  
        System.out.println("Line number 2"); // Again!  
    }  
}
```

# Types

Kinds of values that can be stored and manipulated.

**boolean:** Truth value (**true** or **false**).

**int:** Integer (0, 1, -47).

**double:** Real number (3.14, 1.0, -2.1).

**String:** Text (“hello”, “example”).

# Variables

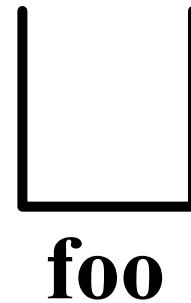
A “box” that stores a value of one type.

Form:

***TYPE NAME;***

Example:

String foo;



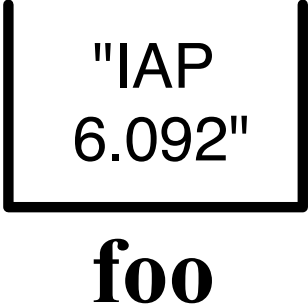
# Assignment

Use = to give variables a value.

Example:

```
String foo;
```

```
foo = "IAP 6.092";
```



The diagram consists of a rectangular box with a thick black border. Inside the box, the text "IAP" is on the top line and "6.092" is on the bottom line, both in a black sans-serif font. Below the box, the word "foo" is written in a bold, black, sans-serif font.

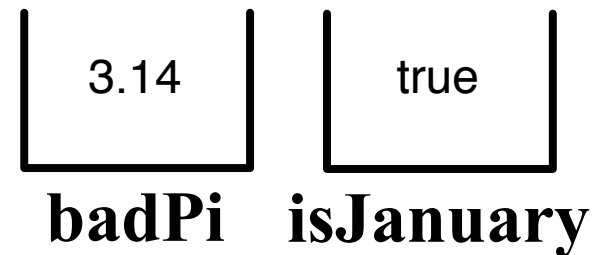
# Assignment

Can be combined with a variable declaration.

Example:

```
double badPi = 3.14;
```

```
boolean isJanuary = true;
```





```
class Hello3 {  
    public static void main(String[] arguments) {  
        String foo = "IAP 6.092";  
        System.out.println(foo);  
        foo = "Something else";  
        System.out.println(foo);  
    }  
}
```

# Operators

Symbols that perform simple computations

Assignment: =

Addition: +

Subtraction: -

Multiplication: \*

Division: /

# Order of Operations

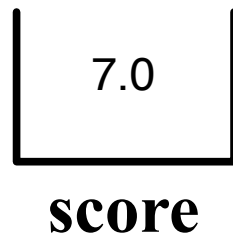
Follows standard math rules:

1. Parentheses
2. Multiplication and division
3. Addition and subtraction

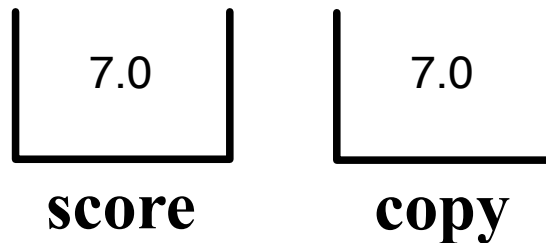
```
class DoMath {  
    public static void main(String[] arguments) {  
        double score = 1.0 + 2.0 * 3.0;  
        System.out.println(score);  
        score = score / 2.0;  
        System.out.println(score);  
    }  
}
```

```
class DoMath2 {  
    public static void main(String[] arguments) {  
        double score = 1.0 + 2.0 * 3.0;  
        System.out.println(score);  
        double copy = score;  
        copy = copy / 2.0;  
        System.out.println(copy);  
        System.out.println(score);  
    }  
}
```

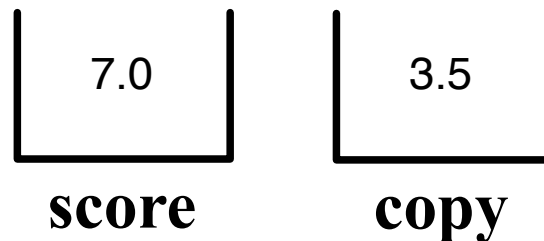
```
class DoMath2 {  
    public static void main(String[] arguments) {  
        double score = 1.0 + 2.0 * 3.0;  
        System.out.println(score);  
        double copy = score;  
        copy = copy / 2.0;  
        System.out.println(copy);  
        System.out.println(score);  
    }  
}
```



```
class DoMath2 {  
    public static void main(String[] arguments) {  
        double score = 1.0 + 2.0 * 3.0;  
        System.out.println(score);  
        double copy = score;  
        copy = copy / 2.0;  
        System.out.println(copy);  
        System.out.println(score);  
    }  
}
```



```
class DoMath2 {  
    public static void main(String[] arguments) {  
        double score = 1.0 + 2.0 * 3.0;  
        System.out.println(score);  
        double copy = score;  
        copy = copy / 2.0;  
        System.out.println(copy);  
        System.out.println(score);  
    }  
}
```





# Division (/)

```
double score = 7.0 / 2.0; // = 3.5
```

What about integers?

# Integer Division

When dividing integers, the fractional part is discarded:

```
int score = 7 / 2;
```

# Assigning Different Types

What happens when assigning different types?

```
double score = 3.5;
```

```
int otherScore = score;
```

# Assigning Different Types

What happens when assigning different types?

```
double score = 3.5;
```

```
// Type mismatch: cannot convert from double  
to int
```

```
int otherScore = score;
```

# Converting Between Types

If a conversion will not “lose” data, it is automatically permitted.

If a conversion might lose data, it must be forced (**casting**):

```
double score = 3.5;
```

```
int otherScore = (int) score;
```

```
double yetAnotherScore = otherScore;
```

# String Concatenation (+)

```
String text = "hello" + " world";
```

```
text = text + " number " + 5; // converted  
                             automatically
```

```
// text = "hello world number 5"
```

# Assignment: GravityCalculator

Compute the position of a falling object:

$$x(t) = 0.5 \times at^2 + v_i t + x_i$$

Details:

<http://stellar.mit.edu/S/course/6/ia11/6.092/>