Identifiers and types

Inigo Montoya
You killed my Father
Prepare to die
Announcements

Homework due next Friday at 6:00pm (submit on moodle)

Will post today
An identifier is the name of a variable (or object, class, method, etc.)

- Case sensitive
- Must use only letters, numbers or _
- Cannot start with a number
- (Some reserved identifiers)
Identifiers

Already did this in week 1!
See: LogicError.java

```java
public class LogicError {

    public static void main(String[] args) {
        double answer = 1/2;
        System.out.println("one divided by two is: ");
        System.out.println(answer);
    }
}
```
Identifiers

Which identifiers are valid?

1) james parker  
2) BoByBoY  
3) x3  
4) 3x  
5) x________  
6) ________x  
7) Home.Class  
8) Five%  
9) x 1
public class FLOAT {

    public static void main(String[] args) {
        float FLoAT, Float, fLoAt, FLOaT, FLOAt, FLOaT, FLOAT, FLOAT;
        FLoAT = 2;
        FLOaT = 5;
        FLOAt = 2;
        Float = 4;
        fLoat = 2;
        FLOAt = (-FLOAt+fLoAT(FLOaT*FLOaT-Float*FLOAT*FLOAT))/(fLoat*FLoAT);
        FLOaT = (-FLOAT-fLoAT(FLOaT*FLOaT-Float*FLOAT*FLOAT))/(fLoat*FLoAT);
        System.out.println(FLOAt);
        System.out.println(FLOaT);
    }
}
Primitive Types

boolean - True or false
char - (character) A letter or number
int - (integer) Whole numbers
long - (long integers) Larger whole numbers
float - Decimal numbers
double - Larger decimal numbers
Many other languages allow conversion between integer and boolean

True = 1 and false = 0

Java does not allow this
int vs long?

**int** - Whole numbers in the approximate range:
-2.14 billion to 2.14 billions ($10^9$)

**long** - Whole numbers in the approximate range:
-9.22 quintillion to 9.22 quintillion ($10^{18}$)

Using **int** is standard (unless you really need more space, for example scientific computing)

(See: IntVsLong.java)
float vs double?
**float vs double?**

**float** is now pretty much obsolete.

**double** takes twice as much space in the computer and 1) has wider range and 2) is more precise

Bottom line: use double (unless for a joke)
float and double

Both stored in scientific notation

```java
double x = 2858291;
```

Computer's perspective:

\[ x = 2.858291 \times 10^6 \]
Welcome to binary

Decimal:  
$1/2 = 0.5$  
$1/3 = 0.3333333$  
$1/10 = 0.1$

Binary:  
$0.1$  
$0.010101010101$  
$0.0001100110011$

double is often just an approximation!
Numerical analysis

Field of study for (reducing) computer error

See: SubtractionError.java

Can happen frequently when solving system of linear equations
int or double?

If you are counting something (money), use `int`

If you are dealing with abstract concepts (physics), use `double`

`int` doesn't make “rounding” mistakes

Non-primitive types exist for both `int` and `double` for more expression/precision
Primitive type hierarchy

int < long < float < double

If multiple primitive types are mixed together in a statement, it will convert to the largest type present.

Otherwise it will not convert type.
Primitive type hierarchy

```
int x;
double y;

x+y
```

Converted to double

```
int x;
int y;

x/y
```

Not converted (still int)
Integer division

See: SimpleDivision.java

Can be fixed by typecasting:

`(double)1/2`