

COMPUTER PROGRAMMING IN JAVA

COLUMBIA UNIVERSITY HIGH SCHOOL SCIENCE HONORS PROGRAM

Basic Java, Part 1
Lecture
2007 Feb 09 Sat

Review

- What is computer science?
 - Essentially, the study of algorithms.
- What is an algorithm?
 - A step-by-step, unambiguous procedure for completing a well defined task.
 - Examples
 - Euclid's algorithm for finding the greatest common denominator
 - Starting a car
 - Driving a manual transmission car
 - Sorting a deck of cards
- How Java works
 - source file
 - text, human-readable, in the Java language
 - class file
 - binary, machine-readable, for the JVM
- Tools
 - PuTTY (Windows)
 - For connecting to CUNIX, the computer environment for students at Columbia
 - WinSCP (Windows)
 - For transferring files back and forth from your user account

Data Types & Variables

- Boolean
 - **boolean**
 - **true, false**

- “Integers”
 - **short**
 - 16 bits
 - Smallest data type for storing whole numbers
 - -32,768 to 32,767
 - **int**
 - 32 bits
 - Standard integer data type
 - **long**
 - 64 bits (big!)
 - Big!
 - -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 (+/- 9 quintillion)
- “Floating point numbers”
 - **float**
 - 32 bits
 - **double**
 - 64 bits
 - Standard floating point data type
- “Strings”
 - **char**
 - 16 bits
 - Unicode
 - **String**
 - An array of characters
 - Examples
 - “hello”
 - “world”
 - Also Unicode
- “Arrays”
 - **int[]**, **boolean[]**, etc
 - 0-indexed!

How do we create variables?

To create a variable, we first have to declare it. Note that we can declare a variable more than once in the same scope.

How to assign variables?

With the assignment operator “=”. This is also known as “initialization”; when you get a compiler error that says “Variable whatever has not been initiated”, this is what it means.

E.g.:

```
int i = 5;  
boolean b = false;
```

Invalid assignments:

```
String s = 1;  
double d = 'z';
```

This is called “type checking”. Java is “strongly typed”.

Comparison Operators

Now that we know the data types, what can we do with them? How about basic true/false statements?

- ==
 - is equal to (note that it’s **TWO** equals signs, not one!)
- !=
 - is not equal to
- >
 - is greater than
- >=
 - is greater than or equal to
- <
 - is less than
- <=
 - is less than or equal to
- **&&**
 - logical AND
- **||**
 - logical OR
- **!**
 - logical NOT

Important: Strings use `.equals()`!

Arithmetic Operators

Two types:

- **Binary**
 - +
 - -
 - *
 - /
 - %
- Promotion rules
 - It's okay to make something “bigger”: widening promotions
 - **int** => **float**
 - **short** => **long**
- It's bad to make something “smaller”: narrowing promotions
 - **double** => **int**
 - **long** => **short**
- String promotions
 - **int, short, long, float, double, char** => **String**
- Unary
 - +
 - unary plus
 - -
 - unary minus
 - ++
 - increment
 - post versus pre
 - --
 - decrement
 - post versus pre

Control Flow

Remember the definition of an algorithm: a step-by-step series of unambiguous instructions. Part of the algorithm might involve doing something conditionally, or doing another thing many times. The features of a programming language that allow us to do this are collectively called “control flow structures”.

if/else if/else

```
if ( <boolean> )
{
    ...
}
else if ( <boolean> )
{
    ...
}
else if ( <boolean> )
{
    ...
}
...
else
{
    ...
}
```

for loop

```
for ( <variable initialization>; <boolean>; <step assignment> )
{
    ...
}
```

while loop

```
while ( <boolean> )
{
    ...
}
```

Note that a semicolon delimits a “statement”. There is no semi-colon after if statements, for loops, or while loops!

Arrays

Looping over an array

```
int[] array = new int[14];
for (int i = 0; i < array.length; i ++)
{
    ...
}

int i = 0;
while (i < array.length)
{
    ...
    i ++;
}
```