CSCI 151 Data Structures - Notes

James Capuder, Oberlin College February 6, 2015

2/4/15 - Intro to Java

Why Java?

- Speed: Compiler vs interpreter
- Platform independence: Java runs on virtual machine, which is the same on every platform (Windows, OSX, Android)
- Can send program without source code
- Memory Management (Garbage Collection)
- Pure object-oriented language
- Strongly Typed Language

Hello World Notes

- File name has to be identical to main class
- Octothorp to comment in python
- double forward slash to line comment in Java
- Block comment, everything between forwardslash star to star forwardslash, even works across multiple lines
- Finally, variation of the block comment, the Javadoc comment, goes from forwardslash doublestar to star forwardslash.

Whitespace

- lines end in ';'
- BLOCKs are surrounded by {}
- strips away all whitespace

Visibility

- Public, anyone can access
- Private, only I can access the variable, along with other variables in the same class
- protected

Variables

- In Java, variables must be declared with a type and visibility

Primitive Types

Types	Description
Byte	8 bit integer
Short	16 bit integer
Int	32 bit integer
Long	64 bit integer
Float	32 bit floating point number
Double	64 bit floating point number
Char	16 bit unicode character
Boolean	true/false

Non-primative types/objects

Objects	Description
String	Character String, double quotes
Arrays	TBC

Operators

Operator	Function
=	assignment
+	addition
-	subtraction
*	multiplication
/	division
%	modulus division
int y = a/b	integer division
f=(double) a/b	cast a into a double

Arrays

- int a[]; - declares a variable a to be an array of integers
- a = newint[4]; - creates an array with 4 slots
- int a[] = { 2,4,8,16 }; - does both of the above steps and assigns values to the indices

Logical Operators

Operator	Function
==	equality
!=	not equal
dobule ampersand	and
||	or
!	not
wedge symbol (shift 6)	exclusive or

Loops and decision points

if statements

```
if (<test>) {
    statement;
    otherstatement;
} else {
    this;
    that;
}
something else;
```

Elif statements

```
if (<test>) {
    statement;
} else if (othertest) {
    otherstatement;
} else {
    lastone;
}
```

for loops

```
for ( <init> ; <test> ; <incr> ) {
    command;
    cmd2;
}
for (int i=0 ; i<10 ; i+=1 ) {
    command;
    cmd2;
}
public static void main(String[] args) {
    for (string s: args) {
```

```
        System.out.println(s);
    }
}
```

pre and post increment operators

```
int a=3;
a++; // a is now 4 (post increment)
++a; // a is now 5 (pre-increment)
int b, c;
a = 5;
b = a++; // a is now 6, b is now 5
a = 5;
b = ++a // a is now 6, b is now 6
```