Logistics

Labs

- Lab 1 extended deadline (Sunday)
- Lab 2 will be an easy quiz
- Go to your normal labs this week, may re-arrange after getting solid attendance numbers

Reading: See schedule

- BJP Ch 1-7
- Lab Manual chapters

Project 1

- Posted, deadline extended (Wednesday 2/3)
- Might field questions later today

Goals Today

- Basic java syntax
- Exercise: Write a static method which returns a reversed copy of a parameter array
By Now Make sure you . . .

- Have a development environment
- Can create new .java files
- Experimented with hello world type programs
- Finished/close to finishing Lab 1, submit to Blackboard
Basic Structure of a Java Source File

public class SomeClass {
    public static typeR myMethod1(typeA a, typeB b) {
        typeC c = some code;
        some more code;
        return someR;
    }

    public static int calls = 0;
    public static int theAverage(int x, int y, int z) {
        int a = x + y + z;
        a = a / 3;
        calls = calls + 1;
        return a;
    }

    public static void main(String[] args) {
        int myAvg = theAverage(1, 2, 3);
        int metaAvg = SomeClass.theAverage(myAvg, 2*myAvg, 3*myAvg);
        System.out.println("The average is " + myAvg);
        System.out.println("The meta average is " + metaAvg);
        System.out.println("Calls to average: " + calls);
        return;  // optional
    }
}
Every Programming Language

Start by looking for the following

- Comments
- Statements/Expressions
- Variable Types
- Assignment
- Basic Input/Output
- Conditionals (if-else)
- Iteration (loops)
- Aggregate data (arrays, structs, objects, etc)
- Function Declarations
- Library System
Demo.java in basic-syntax.zip contains examples for today
Also several other programs in the zip

Note: All code examples are posted some time after class in the same spot as the lecture slides. Where are the lecture slides posted?
Conditionals

- if/else
  - Demo.java
  - Act on a boolean
  - Comparisons: ==, !=, <, >, <=, >=
  - Nesting
  - Chaining

- switch/case
  - Useful in some special cases, but not generally
  - Maybe we’ll talk about it some time
4 flavors

- **Now - Iteration.java**
  - `while`
  - `Traditional for`

- **Maybe Later**
  - `do while`
  - `for each (collections)`
while(
    condition
) {
    this gets done repeatedly;
}
this gets done once;

while(condition)
    this gets done repeatedly;
    as does this;
    and this;
}
this gets done once;

Look at Iteration.java
for

for(initialize; condition; update)
    do some stuff repeatedly;
then do this;

for(initialize; condition; update){
    do some stuff repeatedly;
    and some other stuff repeatedly;
}
then do this;
Do you need both for and while?
Arrays - Multiple of the same kind of thing

See ArrayDemo.java

Define  Now there’s a type bleh, it looks like blah
  ▶  Done for you: part of the java language

Declare  Here is a variable, it’s type is bleh

        int ia[] = new int[3];
        double doubss[] = new double[10];
        boolean [] bools = new boolean[4];

Assign  Element foo of variable bar gets value blip

        ia[0] = 1;
        doubss[2] = 1.2345;
        bools[3] = true;

Access  Retrieve element foo of variable bar

        int i = ia[1];
        double d = doubss[4];
        boolean b = bools[0];
**Length**

Arrays carry their length
It's an int (or long?).

```java
int ia[] = new int[3];
System.out.println(ia.length);
int len = ia.length;
for(int i=0; i<ia.length; i++){
    System.out.print(ia[i]+" ");
}
```

Can't change length

// Compile ERROR
```java
ia.length = 20;
```

Why not?

Can cause runtime errors

```java
ia = new int[5];
ia[10] = 12;
Exception in thread "main"
java.lang.ArrayIndexOutOfBoundsException: 10
at ArrayDemo.main(ArrayDemo.java:23)
```
Exercise: Reverse copy of an Array

Write a static method

```
public static int[] reverseCopy(int[] a)
```

which creates a reverse copy of the array `a` and returns it. You will need to do the following.

- Allocate space for the return array
- Iterate through the array `a` and copy elements to the corresponding positions in the return array
- Return the reverse copy

```java
int arr1[] = {5, 4, 3, 2, 1};
int rev1[] = ReverseArray.reverseCopy(arr1);
for(int i=0; i<rev1.length; i++) {
    System.out.print(rev1[i] + " ");
}
System.out.println();  // Expect: 1 2 3 4 5

int [] arr2 = {2, 4, 6, 8};
int [] rev2 = ReverseArray.reverseCopy(arr2);
for(int i=0; i<rev2.length; i++) {
    System.out.print(rev2[i] + " ");
}
System.out.println();  // Expect: 8 6 4 2
```