

More Java

D. Thiebaut

Lecture 4 — CSC212 Fall 2014

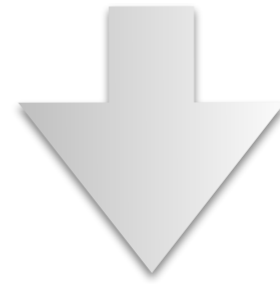
Miscellaneous Java Constructs, and Comments



(What we're not covering today:
Exceptions...)

String[] args in main?

```
class PrintArgs {  
    public static void main(String[] args) {  
        for (int i = 0; i < args.length; i++)  
            System.out.println(String.format("args[%d] = %s", i, args[i]));  
    }  
}
```



```
[beowulf2]
[06:48:27] ~/public_html/classes/212$: javac PrintArgs.java
[beowulf2]
[06:48:31] ~/public_html/classes/212$: java PrintArgs
[beowulf2]
[06:48:35] ~/public_html/classes/212$: java PrintArgs Hello
args[0] = Hello
[beowulf2]
[06:48:40] ~/public_html/classes/212$: java PrintArgs Hello there! 1 2 3 4
args[0] = Hello
args[1] = there!
args[2] = 1
args[3] = 2
args[4] = 3
args[5] = 4
```

```
[beowulf2]
[06:48:40] ~/public_html/classes/212$: java PrintArgs Hello there! 1 2 3 4
args[0] = Hello
args[1] = there!
args[2] = 1
args[3] = 2
args[4] = 3
args[5] = 4
```



Every args[i] is a string!

```
// convert string argument to integer
int someNumber = Integer.parseInt( args[0] );
```

In case you're curious...

```
public static int strToInt( String str ){
    int i = 0;
    int num = 0;
    boolean isNeg = false;

    //check for negative sign; if it's there, set the isNeg flag
    if( str.charAt(0) == '-' ) {
        isNeg = true;
        i = 1;
    }

    //process each char of the string;
    while( i < str.length() ) {
        num *= 10;
        num += str.charAt(i++) - '0'; //minus the ASCII code of '0'
                                     // to get the value of the charAt(i++)
    }

    if (isNeg)
        num = -num;
    return num;
}
```

```
[06:48:40] ~/public_html/classes/212$: java PrintArgs Hello there! 1 2 3 4
```

```
class PrintArgs {  
    public static void main(String[] args) {  
        for (int i = 0; i < args.length; i++)  
            System.out.println(String.format("args[%d] = %s", i, args[i]));  
    }  
}
```



Question: Who is passing "Hello there! 1 2 3 4"?

Implications

- We can pass "**run-time**" information to our programs!
 - names of **files** to work with
 - **switches**:
 - "-q" for "quiet", no output
 - "-d", or "-debug" to make program output debugging information
 - ...
 - can you think of others?

Arrays vs. Strings

- Get the length of an Array

```
int[] A = {1, 1, 2, 3, 5, 8};  
int len = A.length; // 6
```

- Get the length of a String

```
String Name = "Hello!";  
int len = Name.length(); // 6
```

Array Indexes

- **NO NEGATIVE INDEXES!**

Python:

```
A = [1, 1, 2, 3, 5, 8]
print( A[-1] )      # 8
print( A[-2] )      # 5
```

Java:

```
int[] A = {1, 1, 2, 3, 5, 8};
System.out.println( A[A.length - 1] ); // 8
System.out.println( A[A.length - 2] ); // 5
```

User Input

```
import java.util.Scanner;
```

```
class GetInputFromUser {  
  public static void main(String args[]) {  
    int a;    float b;    String s;  
    Scanner in = new Scanner(System.in);  
  
    System.out.println("Enter a string >");  
    s = in.nextLine();  
    System.out.println("You entered: "+s);  
  
    System.out.println("Enter an integer >");  
    a = in.nextInt();  
    System.out.println("You entered: "+a);  
  
    System.out.println("Enter a float >");  
    b = in.nextFloat();  
    System.out.println("You entered: "+b);  
  }  
}
```

Difference between
"***Super***" & "***This***"
Keywords

```

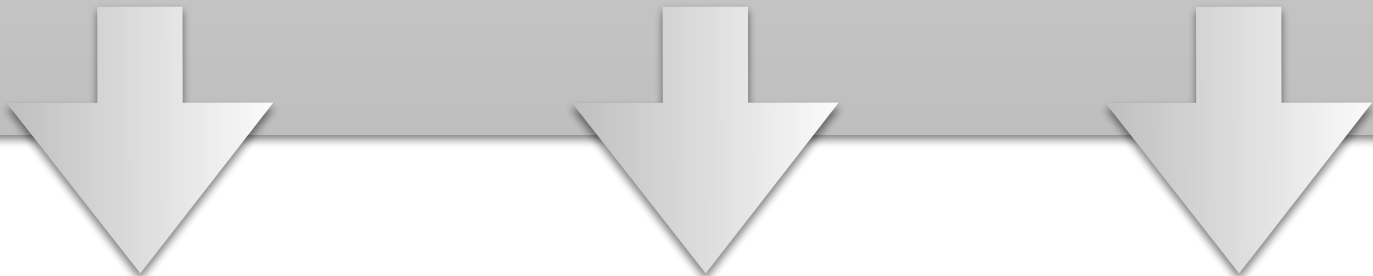
public class ThisSuperExample{
    String name = "";
    static int Id = 0;           // only 1 in program, no matter how many objects
    int currentId;             // each object has an Id
    int age;

    public ThisSuperExample( String n, int x ) {
        currentId = Id++;      // first object gets 0, second gets 1, etc.
        name = n;
        age = x;
    }

    public void display() {
        System.out.println( name + " (" + currentId + ") " + age );
    }

    static public void main( String[] args ) {
        ThisSuperExample x = new ThisSuperExample( "Alka", 3 );
        ThisSuperExample y = new ThisSuperExample( "Romeo", 4 );
        x.display();
        y.display();
    }
}

```



Alka (0) 3
Romeo (1) 4

Question: Overload constructor for setting age to 0, automatically

```
public class ThisSuperExample{
    String name = "";
    static int Id = 0; // only 1 in program, no matter how many objects
    int currentId; // each object has an Id
    int age;

    public ThisSuperExample( String n, int x ) {
        currentId = Id++; // first object gets 0, second gets 1, etc.
        name = n;
        age = x;
    }

    public ThisSuperExample( String n ) {
        // ???????????
    }

    public void display() { ... } // collapsed

    static public void main( String[] args ) { ... } // collapsed
}
```

Answer 1: Add new statements


```
public class ThisSuperExample{
    String name = "";
    static int Id = 0; // only 1 in program, no matter how many objects
    int currentId; // each object has an Id
    int age;

    public ThisSuperExample( String n, int x ) {
        currentId = Id++; // first object gets 0, second gets 1, etc.
        name = n;
        age = x;
    }

    public ThisSuperExample( String n ) {
        currentId = Id++; name = n; age = 0;
    }

    public void display() { ... } // collapsed

    static public void main( String[] args ) { ... } // collapsed
}
```



But what if the first
constructor is
several pages
long?



Answer 2: call the first constructor!


```
public class ThisSuperExample{
    String name = "";
    static int Id = 0; // only 1 in program, no matter how many objects
    int currentId; // each object has an Id
    int age;

    public ThisSuperExample( String n, int x ) {
        currentId = Id++; // first object gets 0, second gets 1, etc.
        name = n;
        age = x;
    }

    public ThisSuperExample( String n ) {
        this( n, 0 );
    }

    public void display() { ... } // collapsed

    static public void main( String[] args ) { ... } // collapsed
}
```



Introduction to Data Structures

Enumerate All
the methods and
operations possible
with Python Lists



Exercise: Write a Java Class that Implements a Python-Like List