

Java Primer I

CMSC 202

Fall 2012

Java Basics to Learn on Your Own*

- Legal identifier formats
- Variable and constant naming conventions
- Arithmetic operators
- Binary operators
- Unary operators
- Ternary operator
- Logical operators
- Relational operators
- Pre and post-increment/decrement operators
- Operator precedence
- if-then, if-then-else, while

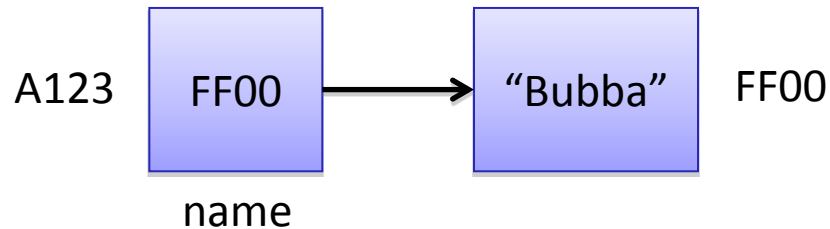
*A link to the spring 2012 slides is on the 202 web site's main page under Recent Announcements. These topics may be found there. And don't forget Google!

Java Data Types

Primitive



Reference



**More later on
reference
variables!**

Java Primitive Types

Type Name	Type of Value	Memory Used	Value Range
byte	integer	1 byte	-128 to 127
short	integer	2 bytes	-32768 to 32767
int	integer	4 bytes	-2147483648 to 2147483647
long	integer	8 bytes	-9223372036854775808 to 9223372036854775807
float	floating point	4 bytes	$-3.40282347 \times 10^{+38}$ to $-1.40239846 \times 10^{-45}$
double	floating point	8 bytes	$+1.76769313486231570 \times 10^{+308}$ to $+4.94065645841246544 \times 10^{-324}$
boolean	true, false	1 byte	not applicable
char	single character	2 bytes	all Unicode characters

Declaring and Using Variables (Basics)

Python

```
salary = 25000.0
age = 32
name = 'Joe'
print(age)
print(myAge)
```

```
linux3[9]% python declarations.py
32
Traceback (most recent call last):
  File "declarations.py", line 5, in <module>
    print (myAge)
NameError: name 'myAge' is not defined
```

Java*

(snippet)

```
double salary = 25000.0;
int age ;
age = 32;
String name = "Joe";
System.out.println(age);
System.out.println(myage);
```

```
linux3[28]% javac Declarations.java
Declarations.java:9: cannot find symbol
symbol : variable myAge
location: class Declarations
    System.out.println(myAge);
                        ^
```

*no global variables allowed in Java

Printing to the Screen

- Unformatted Output

- `System.out.print(...);` leaves cursor on same line
- `System.out.println(...);` cursor moves to next line

- Example Snippet:

```
System.out.print("Hello");  
System.out.print(" there");  
System.out.println("Hello");  
System.out.println(" there");
```

Output:

```
Hello thereHello  
there
```

Printing to the Screen (con't)

- Formatted Output

`System.out.printf(format-string, parameter1, ..., parametern);`

- Example Snippet

```
System.out.printf("Printing an integer: %d%n", 5);
```

```
System.out.printf("%d %c %5.2f", 1, 'a', 2.123);
```

```
String word = "Hello";
```

```
System.out.printf("%s", word);
```

Output:

Printing an integer: 5

1 a 2.12Hello

- Reference

- Java API, PrintStream class, printf method

A First Java Program (using GL)

```
public class HelloWorld {  
    public static void main(String[ ] args){  
        System.out.println("Hello, world!");  
    }  
}
```

```
linux3[95]% ls
```

```
Declarations.java HelloWorld2.java HelloWorld.java
```

```
linux3[96]% javac HelloWorld.java
```

```
linux3[97]% ls
```

```
Declarations.java HelloWorld2.java HelloWorld.class HelloWorld.java
```

```
linux3[98]% java HelloWorld
```

```
Hello, world!
```


More on Data Types

- ***Simple*** (integral, scalar)
 - Contains only one value at any given time
 - Java Examples:

```
int x = 5;    float e = 2.71;    char initial = 'S';
```
- ***Complex*** (aggregate, composite)
 - May contain one or more values
 - ***Heterogeneous*** or ***homogenous***
 - Examples:
 - A ***list*** in Python (heterogeneous)
 - An ***array*** in Java (homogeneous)

Java Arrays

- Homogeneous
- Reference type
- Declaration Format:

<data-type>[] <variable-name>;

- Example Declarations:

int scores[]; char grades[];



So, what's in memory?

Java Arrays (con't)

```
public class ArraysOneDim {  
    public static void main(String[] args) {  
        int scores[];  
        char grades[];  
        System.out.println("scores: ", scores);  
        System.out.println("grades: ", grades);  
    }  
}
```

ArraysOneDim.java

```
linux2[13]% javac ArraysOneDim.java  
ArraysOneDim.java:6: cannot find symbol  
symbol   : method println(java.lang.String,int[])  
location: class java.io.PrintStream
```

Compiling on GL

```
    System.out.println("scores: ", scores);  
        ^
```

```
ArraysOneDim.java:7: cannot find symbol  
symbol   : method println(java.lang.String,char[])  
location: class java.io.PrintStream
```

```
    System.out.println("grades: ", grades);  
        ^
```

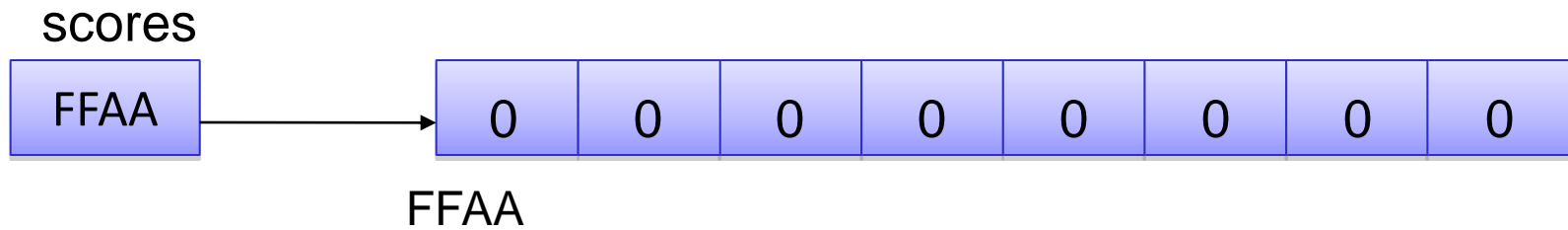
Java Arrays (con't)

- Initializing an Array

<data-type>[] *<variable-name>* =
new *<data-type>*[*number-of-elements*];

- Example:

int scores[] = new int[8]; OR
int scores[]; scores = new int[8];

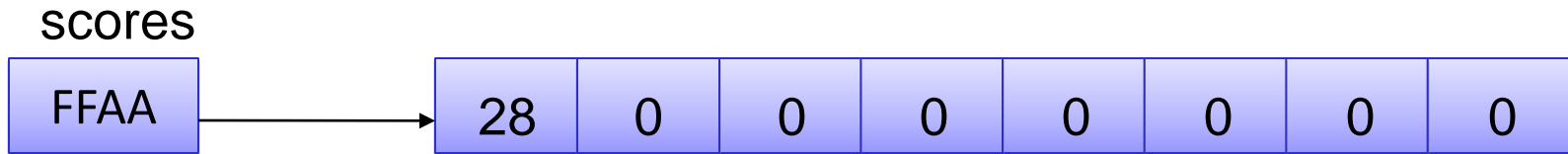


Java Arrays (con't)

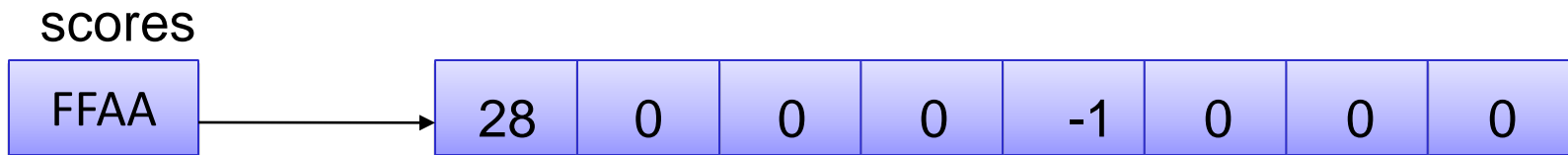
Data Type	Default Value (for elements)
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
String (or any reference)	null
boolean	false

Java Arrays (con't)

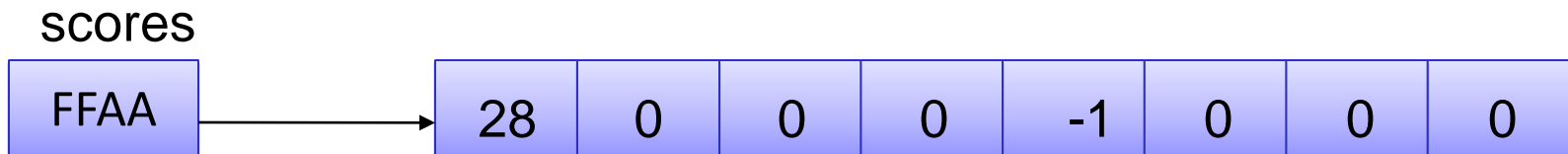
```
scores[0] = 28;
```



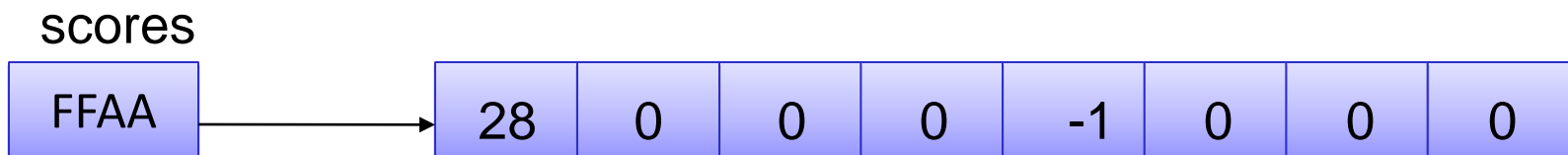
```
scores[4] = -1;
```



```
System.out.print(scores[3]);
```



```
System.out.print(scores.length);
```



Multi-dimensional Arrays

- Two-dimensional Array Declaration:

<data-type>[] [] *<variable-name>*;

- Example Declaration:

```
char [ ] [ ] ticTacToeBoard;
```

- Initializing:

```
ticTacToeBoard = new char[3][3]; OR
```

```
three = 3;
```

```
ticTacToeBoard = new char[three][three];
```

	[0]	[1]	[2]
[0]	"	"	"
[1]	"	"	"
[2]	"	"	"

Multi-dimensional Arrays (con't)

```
ticTacToeBoard[0][2] = 'X';
```

	[0]	[1]	[2]
[0]	"	"	'X'
[1]	"	"	"
[2]	"	"	"

```
char move = 'O';
```

```
ticTacToeBoard[1][1] = move;
```

	[0]	[1]	[2]
[0]	"	"	'X'
[1]	"	'O'	"
[2]	"	"	"

```
System.out.print(  
ticTacToeBoard[0][0]);
```

	[0]	[1]	[2]
[0]	"	"	'X'
[1]	"	'O'	"
[2]	"	"	"

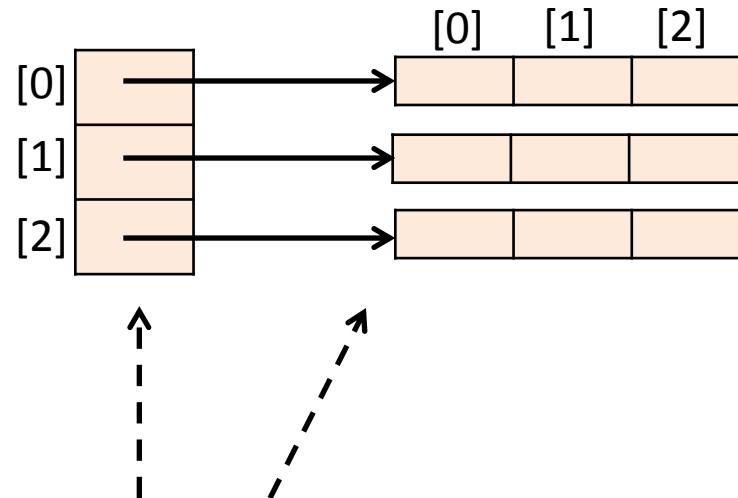
```
int square1 = 2, square2 = 1;
```

```
ticTacToeBoard[square1][square2] = 'O';
```

	[0]	[1]	[2]
[0]	"	"	'X'
[1]	"	'O'	"
[2]	"	'O'	"

Multi-dimensional Arrays (con't)

Best to think of a two-dimensional array as an “array of arrays”



Each contains the
memory address of it's
corresponding row