

COMS W1114 - Java Lab

Lab 2
Thursday, February 5, 2004

Note

- Reading:
 - Theory: Ch 0, 5.1-5.3, 1.1-1.6
 - Programming: Ch 1, Ch 2, 3.1
- Additional resource:
 - Bruce Eckel's "Thinking in Java, 3rd Edition". Free online.
<http://www.mindview.net/Books/TIJ/>
- HW1 due Thursday, February 12 at 5p
 - submit theory to Janak (he'll go over "how" in class)
 - submit programming online (we'll review how today)

What we are covering today

- Quick review of lab 1
- Data types and Conversion
- Arrays
- Command line input
- Output
- Math operators
- How to submit your programming assignment (submit script)

Quick review of last lab

- Unix connections
 - all ok? Everyone logged in? If not, go ahead.
 - Everyone have HelloWorld.java program?
- HelloWorld.java structure. Compile review. Run review.
- Comments, Classes, and Methods
- Variables, Statements, Expressions
- Errors:
 - syntax and semantic
 - compile time vs. runtime

Variables

- What are they? A place to store information so you, the programmer, can accomplish a task.
- They have names that you give them - so you can refer to them directly.
- They have a type (datatype); a set of rules about what they represent and how they represent it.

Data types

- Primitive data types vs. “full blown” objects
- Primitives:
 - int, double, float, long, boolean, char
 - declaration simple:

```
datatype variableName [= value](optional);
double insideTemp; //not initialized
int myFavoriteNumber = 27; //initialized
```
 - use simple. Just use it by name.
 - What if you try to use it and it is not initialized?
- Full blown objects. (What is that?!)
 - Objects - you’ll tackle in theory lecture, but...

```
String myString = new String();
```

Data conversion (1)

- When Object is declared as one type, it cannot always be used everywhere

```
String myString = new String("234");  
int remainder = myString % 2; (ERROR)  
(MOD operator does not work with a String)
```

- Or, when it is used, it doesn't work as you'd hope:

```
int price = 6.8; // price is 6, not 6.8
```

Try it.

Data conversion (2)

- Depending on the Object, you can *convert* your data or *cast* it to a different type.

- Tells compiler to promote your class to one that you define.

- Some conversion "automatic"

```
System.out.println("My favorite number is " + myFavoriteNumber);
```

How? via objects `toString()` method

- other conversions are not automatic: prior String example:

```
int remainder = myString % 2; (ERROR)
```

- What happens when you need an int, and you have myString?
(String objects do not have a `toInteger()` method.)

- Use Integer *Object* to do the work for you:

```
int temp = Integer.parseInt(string);
```

- Try it!

Type Casting

- When you have an object of one type and you want to use it (cast it) as another type.

- Motivation:

```
float kilos;  
int est = 45;  
kilos = est * 1.2;
```

What happens? Compile error. Try it.

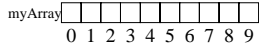
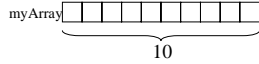
- Solve with a cast:

```
float kilos;  
int est = 45;  
kilos = (float) (est * 1.2);
```

- Typical problem when doing division.

Arrays

- A collection of data types (all the same type)
`int[] myArray = new int[10];`
- Easy to visualize array
- Array has length (here 10)
- Indexing = identifying a specific cell of the array
- Index `0..length-1`
- Access with `myArray[index]`
- (note: some languages use `1..N`)
- Trying to grab cell outside the array bounds causes an error (runtime)
- Treat a cell like a variable. (for assignment. for access. etc.)



Arrays (continued)

Try it:

```
int[] myArray = new int[5];
myArray[0] = 10;
myArray[1] = 20;
myArray[2] = 30;
myArray[3] = myFavoriteNumber;
myArray[4] = 50;
System.out.println("The value in the third
cell of myArray is " + myArray[2]);
```

Command line input

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

- `String[] args` look familiar?
Yes! Its's an array!
 - Access it just like an array you defined
 - What are the *bounds* of the array?
- For our purposes you can *hard code* the access since we know we want the first two (bad form):
`args[0]` and `args[1]` are the first two arguments from the command line.

Output

- Printing to standard out (stdout)
 - `System.out.println();`
 - `System.out.print();`
- What's the difference?
- Try it.

Math Operators (in Gently Java - pp 40)

- binary operators
 - + - * / %
- unary operators
 - ! ++ -- += -= (others)
- relational operators
 - == <= >=

Submit your program

1. Make sure you have only the files you want to send us in a directory (organize your work)
2. Change to this directory
3. Run the script
 - `~cs1004/bin/submit`
4. You are prompted to verify the files to submit. If ok, say "Yes".
5. Look for an email to verify we received it.

Feel free to submit many times before the deadline. If after deadline, we will see it; please let us know when you are taking late days.

Wrap up

- HW1 by 5p 2/12/04 unless you are using late days.
- Variations of these slides are posted this afternoon (typically always after class; not before)
- Reminder: my OH M 1:30-3:30 in 251 ET
