

Objectives

- More Java fundamentals
 - Static typing
 - Arithmetic, shortcuts
 - `java.lang` classes: `Math` and `String` class
 - Control Structures
 - Arrays

Sept 14, 2011

Sprenkle - CSCI209

1

Review: Assign 0

- How did it go?
 - How long did it take?
- My conventions

Sept 14, 2011

Sprenkle - CSCI209

2

Review

- What are some of the primitive data types of Java?
- What is the syntax for declaring a variable in Java?

Sept 14, 2011

Sprenkle - CSCI209

3

Review: Python Transition **Warning**

You cannot redeclare a variable name in the same scope

- OK:

```
int x = 3;
x = -3;
```

- Not OK:

```
int x = 3;
int x = -3;
...
boolean x = true;
```

Compiler errors

Sept 14, 2011

Sprenkle - CSCI209

4

Constants

- Read-only variables
 - Cannot be assigned new values
- Keyword `final` precedes data type
 - Example within a method:

```
final double CM_PER_INCH = 2.540;
```

Why might we want to use constants *within a method*?

Sept 14, 2011

Sprenkle - CSCI209

5

Class Constants

- Constant variable for all methods in class or for multiple classes
 - Much more common than constant instance variables
- Requires `static` keyword
 - `static`: "for class"
 - Also used for methods (will see more later)

```
static final double CM_PER_INCH = 2.540;
```

Sept 14, 2011

Sprenkle - CSCI209

6

Arithmetic, Relational Operators

- Java has most of the same operators as Python:
 - Arithmetic operators: +, -, *, /, %
 - **No power operator:** **
 - Relational operators: ==, !=, <, >, <=, >=
 - Evaluate to a **boolean** value
 - Increment and decrement
 - += x, -= y, etc.
 - Additional shortcut for += 1, -= 1: ++, --

Sept 14, 2011

Sprenkle - CSCI209 `Conversion.java` 7

INTRO TO JAVA LIBRARIES

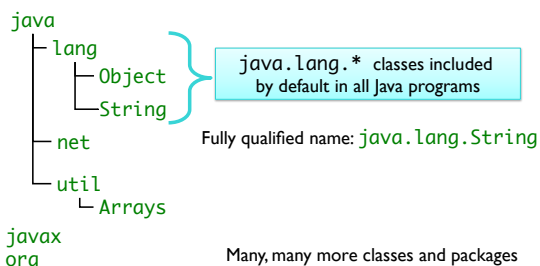
Sept 14, 2011

Sprenkle - CSCI209

8

Java Libraries

- Organized into a hierarchy of **packages**



Sept 14, 2011

Sprenkle - CSCI209

9

Java API Documentation

- API:** Application Programming Interface
 - What the class can do for YOU!
 - Complete documentation of every class included with the JDK
 - Every method and variable contained in class
- <http://download.oracle.com/javase/6/docs/api>
- Bookmark it!
 - Too many classes, methods to remember them all
 - Refer to it often

Sept 14, 2011

Sprenkle - CSCI209

10

java.lang.Math class

- Similar to Python's math module
- Included by default** in every Java program
- Contains useful mathematical functions (methods) and constants (fields):
- Look at `java.lang.Math` API online
 - <http://download.oracle.com/javase/6/docs/api/>
 - Note how API is specified

Sept 14, 2011

Sprenkle - CSCI209

11

java.lang.Math class

- Example Uses:

```

double y = Math.pow(x, a);
double z = Math.sin(y);
double d = Math.exp(4.59) * Math.PI;
  
```

Use `Classname.methodname()` to call Math's methods because they're **static**

`MathExample.java`

Sept 14, 2011

Sprenkle - CSCI209

12

java.lang.String class

- Similar functionality to Python but different ways to use
- Strings are represented by **double quotes**:
""
- Single quotes represent **chars** only
- Examples:

```
String emptyString = "";
String niceGreeting = "Hello there.";
String badGreeting = "What do you want?";
```

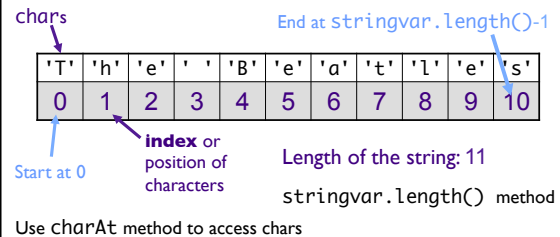
Sept 14, 2011

Sprenkle - CSCI209

13

Strings

- A **char** at each position of String
stringvar = "The Beatles"



Sept 14, 2011

Sprenkle - CSCI209

14

String methods: substring

- Like *slicing* in Python
- String substring(int beginIndex)
➤ Returns a new String that is a substring of this string, from beginIndex to end of this string
- String substring(int beginIndex, int endIndex)
➤ Returns a new String that is a substring of this string, from beginIndex to endIndex

```
String greeting = "Hello, Clark Kent!";
String subStr = greeting.substring(7);
String subStr2 = greeting.substring(7, 12);
```

Python Gotcha: Can't use negative numbers for indices as in Python

String Concatenation

- Use + operator to concatenate Strings

```
String niceGreeting = "Hello";
String firstName = "Clark";
String lastName = "Kent";
String blankSpace = " ";

String greeting = niceGreeting + "," +
    blankSpace + firstName +
    blankSpace + lastName;

System.out.println(greeting);
```

Prints "Hello, Clark Kent"

Sept 14, 2011

Sprenkle - CSCI209

16

String Concatenation

- If a String is concatenated with something that is not a String, the other variable is converted to a String automatically.

```
int totalPoints = 110;
int earnedPoints = 87;
float testScore = (float) earnedPoints/totalPoints;

System.out.println("Your score is " + testScore);
```

Converted to a String

Sept 14, 2011

Sprenkle - CSCI209

17

StringBuffers vs Strings

- Strings are "read-only" or *immutable*
➤ Same as Python
- Use StringBuffer to manipulate a String
- Instead of creating a new String using
➤ String str = prevStr + " more!";
- Use **new** keyword: allocate memory to a new object
StringBuffer str = new StringBuffer(prevStr);
str.append(" more!");
- Many StringBuffer methods
➤ toString() to get the resultant string back

Sept 14, 2011

Sprenkle - CSCI209

18

String Comparison: equals

- **boolean** equals(Object anObject)

- Compares this string to the specified object

```
String string1 = "Hello";
String string2 = "hello";
boolean test;
test = string1.equals(string2);
```

- **test** is false because the Strings contain different values

Sept 14, 2011

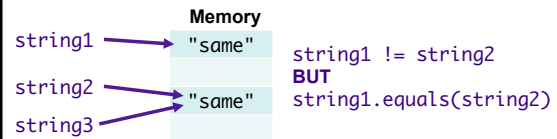
Sprenkle - CSCI209

19

Python Gotcha: String Comparisons

- `string1 == string2` will **not** yield the same result as `string1.equals(string2)`

- `==` tests if the **objects** are the same
 - **not** if the **contents** of the objects are the same
- Similar to `is` operator in Python



Sept 14, 2011

Sprenkle - CSCI209

Equals.java

20

String methods: and many more!

- **boolean** endsWith(String suffix)
- **boolean** startsWith(String prefix)
- **int** length()
- String toLowerCase()
- String trim(): remove trailing and leading white space
- ...
- See `java.lang.String` API for all

Sept 14, 2011

Sprenkle - CSCI209

21

CONTROL STRUCTURES

Sept 14, 2011

Sprenkle - CSCI209

22

Review

- What is the syntax of a *conditional statement* in Python?

Sept 14, 2011

Sprenkle - CSCI209

23

Control Flow: Conditional Statements

- **if** statement

- **Condition** must be surrounded by `()`
- Condition must evaluate to a **boolean**
- Body is enclosed by `{ }` if multiple statements

```
if (purchaseAmount < availCredit) {
    System.out.println("Approved");
    availableCredit -= purchaseAmount;
}
else
    System.out.println("Denied");
```

Don't need `{ }` if only one statement in the body
Best practice: use `{ }`

Sept 14, 2011

Control Flow: Conditional Statements

• if statement

```

if (purchaseAmount < availCredit) {
    System.out.println("Approved");
    availableCredit -= purchaseAmount;
}
else
    System.out.println("Denied");
  
```

Condition

Block of code

- Everything between { } is a block of code
 - Has an associated scope

Sept 14, 2011

Sprenkle - CSCI209

25

Scoping Issues

- Everything between { } is a block of code
 - Has an associated scope

```

if (purchaseAmount < availableCredit) {
    availableCredit -= purchaseAmount;
    boolean approved = true;
}
if( ! approved )
    System.out.println("Denied");
  
```

Out of scope
Will get a compiler error

How do we fix this code?

Sept 14, 2011

Sprenkle - CSCI209

26

Fixed

- Move `approved` outside of the `if` statement

```

boolean approved = false;
if (purchaseAmount < availableCredit) {
    availableCredit -= purchaseAmount;
    approved = true;
}
if( ! approved )
    System.out.println("Denied");
  
```

Sept 14, 2011

Sprenkle - CSCI209

27

Logical Operators

Operation	Python	Java
AND		&&
OR		
NOT		!

Sept 14, 2011

Sprenkle - CSCI209

28

Logical Operators

Operation	Python	Java
AND	and	&&
OR	or	
NOT	not	!

Sept 14, 2011

Sprenkle - CSCI209

29

Control Flow: else if

- In Python, was `elif`

```

if ( x%2 == 0 ) {
    System.out.println("Value is even.");
}
else if ( x%3 == 0 ) {
    System.out.println("Value is divisible by 3.");
}
else {
    System.out.println("Value isn't divisible by 2 or 3.");
}
  
```

What output put do we get if x is 9, 13, and 6?

Sept 14, 2011

Sprenkle - CSCI209

30

Control Flow: switch statement

- Like a big if/else if statement
- Works with variables with datatypes **byte**, **short**, **char**, and **int** (And String in Java 7)

```
int x = 3;
switch(x) {
    case 1:
        System.out.println("It's a 1.");
        break;
    case 2:
        System.out.println("It's a 2.");
        break;
    default:
        System.out.println("Not a 1 or 2.");
}
```

ε }

Control Flow: switch statement

```
switch(grade) {
    case 'a':
    case 'A':
        System.out.println("Congrats!");
        break;
    case 'b':
    case 'B':
        System.out.println("Not too shabby!");
        break;
    // Handle c, d, and f ...
    default:
        System.out.println("Error: not a grade");
}
```

Sept 14, 2011

Sprengle - CSCI209

Grades.java

32

Control Flow: while Loops

- while** loop
 - Condition must be enclosed in parentheses
 - Body of loop must be enclosed in {} if multiple statements

```
int counter = 0;
while (counter < 10) {
    System.out.println(counter);
    counter++;
}
System.out.println("Done: " + counter);
```

shortcut

Sept 14, 2011

Sprengle - CSCI209

33

Changing control flow: break

- Exits the current loop

```
while ( <readingdata> ) {
    ...
    if( data == null ) { // shouldn't happen
        break;
    }
}
```

Sept 14, 2011

Sprengle - CSCI209

34

Review

- How do you write a **for** loop in Python?

Sept 14, 2011

Sprengle - CSCI209

35

Control Flow: for Loop

- Very different syntax from Python
- Syntax:

```
for (<init>; <condition>; <execution_expr>)
```

Loop's counter variable,
Usually used in condition

Executed at end of each iteration.
Typically increments or
decrements counter

Sept 14, 2011

Sprengle - CSCI209

36

Control Flow: for Loop Example

```
System.out.println("Counting down...");
for (int count=10; count >= 1; count--) {
    System.out.println(count);
}
System.out.println("Blastoff!");
```

shortcut

- What is the counter variable?
- What is the condition?
- What is the output?
- How written in Python?

Sept 14, 2011

Sprenkle - CSCI209

37

ARRAYS

Sept 14, 2011

Sprenkle - CSCI209

38

Python Lists → Java Arrays

- A Java **array** is like a **fixed-length** list
- To declare an array of integers:
 - `int[] arrayOfInts;`
 - Declaration only makes a variable named `arrayOfInts`
 - Does not initialize array or allocate memory for the elements
- To declare and **initialize** array of integers:
 - `int[] arrayOfInts = new int[100];`

new keyword:

allocate memory to a new object

Sept 14, 2011

Sprenkle

Array Initialization

- Initialize an array at its declaration:
 - `int[] fibNumbs = {1, 1, 2, 3, 5, 8, 13};`

Value	1	1	2	3	5	8	13
Position/index	0	1	2	3	4	5	6

- Note that we do not use the **new** keyword when allocating and initializing an array in this manner
- `fibNumbs` has length 7

Sept 14, 2011

Sprenkle - CSCI209

40

Array Access

- Access a value in a array as in Python:
 - `fibNums[0]`
 - `fibNums[x] = fibNums[x-1] + fibNums[x-2]`
- Unlike in Python, cannot use negative numbers to index items

Sept 14, 2011

Sprenkle - CSCI209

41

Array Length

- All array variables have a **field** called **length**
 - Note: no parentheses because not a method

```
int[] array = new int[10];
for (int i = 0; i < array.length; i++) {
    array[i] = i*2;
}

for (int i = array.length-1; i >= 0; i--) {
    System.out.println(array[i]);
}
```

Sept 14, 2011

Sprenkle - CSCI209 ArrayLength.java 42

Overstepping Array Length

- Java safeguards against overstepping length of array
 - Runtime Exception: "Array index out of bounds"
 - More on exceptions later...
- Example


```
int[] array = new int[100];
```

 - Attempts to access or write to index < 0 or index >= array.length (100) will generate exception

Sept 14, 2011

Sprenkle - CSCI209

43

Command-Line Arguments

- Similar to Python's `sys` module

```
# Make sure there are sufficient arguments.
if len(sys.argv) < 2:
    print "Error: invalid number of command-line arguments"
    print "Usage: python", sys.argv[0], "<filename>"
    sys.exit(1)
```

Contains the command-line arguments

```
public static void main(String[] args) {
    if( args.length < 1 ) {
        System.out.println("Error: invalid number of arguments");
        System.out.println("Usage: java MyProgram <filename>");
        System.exit(1);
    }
}
```

Command-Line Arguments

- `sys.argv[0]` represented name of Python program
- Not same in Java
 - Command-line arguments do not include the

```
# Make sure there are sufficient arguments.
if len(sys.argv) < 2:
    print "Error: invalid number of command-line arguments"
    print "Usage: python", sys.argv[0], "<filename>"
    sys.exit(1)
```

Have to specify program name in Java, e.g.,

```
System.out.println("Usage: java MyProgram <filename>");
```

Sept 14, 2011

Sprenkle - CSCI209

45

Control Flow: foreach Loop

- Introduced in Java 1.5
 - Sun calls "enhanced for" loop
- Iterate over all elements in an array (or Collection)
 - Similar to Python's `for` loop

```
int[] a;
int result = 0;
for (int i : a) {
    result += i;
}
```

for each int element *i* in the array *a*
The loop body is visited once for each element of *a*.

Sept 14, 2011

Sprenkle - CSCI209

46

Arrays

- Assigning one array variable to another
 - ➔ both variables refer to the same array
 - Similar to Python
 - Draw picture

```
int[] fibNums = {1, 1, 2, 3, 5, 8, 13};
int[] otherFibNums;

otherFibNums = fibNums;
otherFibNums[2] = 99;

System.out.println(otherFibNums[2]);
System.out.println(fibNums[2]);
```

fibNums[2] and otherFibNums[2] are both equal to 99

Sept 14, 2011

Sprenkle - CSCI209

47

java.util.Arrays

- `Arrays` is a class in `java.util`
- Methods for sorting, searching, `deepEquals`, fill arrays
- To use class, need `import` statement
 - Goes at top of program, before class definition

```
import java.util.Arrays;
```

ArraysExample.java

Sept 14, 2011

Sprenkle - CSCI209

48

Array Copying

- Copy an array (element-by-element) using the `arraycopy` method in the `System` class

```
System.arraycopy(from, fromIndex, to, toIndex, count);
```

- For example:

```
int[] fibNums = {1, 1, 2, 3, 5, 8, 13};
int[] otherFibNums = new int[fibNums.length];
System.arraycopy(fibNums, 0, otherFibNums, 0, fibNums.length);
otherFibNums[2] = 99;
System.out.println(otherFibNums[2]);
System.out.println(fibNums[2]);
```

```
fibNums[2] = 2,
otherFibNums[2] = 99
```

Sept 14, 2011

Sprenkle - CSCI209

49

To Do

- Assign 1
 - Part 1: Fixing compiler and logic errors from program
 - Part 2: Writing a program to compute a gymnastics score at the Olympics
 - Part 3: Reverse a string
 - EC opportunity
 - Due Friday before class
- Extra credit opportunity
 - Sylvia Earle's lunch (noon, Women's Resource Room) and talk (5:30 p.m., Stackhouse) on Monday

Sept 14, 2011

Sprenkle - CSCI209

50

Python Dictionaries → Java HashMaps

- We'll discuss later

Sept 14, 2011

Sprenkle - CSCI209

51

Benefits of Static Typing

- Look at `dynamic_typing.py`
- Discussion questions
 - What is the type of `data` at the end of the program?
 - How difficult is this program to understand?
 - If you had to debug this program, how easy/difficult would it be?
 - What is a benefit of dynamic typing?

`alternative_dynamic_typing.py`

Sept 14, 2011

Sprenkle - CSCI209

52

Benefits of Static Typing

- Easier to remember type of variable
 - Know operations that can be executed on a variable of a certain type
- Compiler can check that you're only using valid operations for this type
- More benefits later this semester

Sept 14, 2011

Sprenkle - CSCI209

53

More Why Java?

- More **structure** emphasizes/requires better **design**

Sept 14, 2011

Sprenkle - CSCI209

54

Python to Java Gotchas

- Every variable needs to be declared before it is used
- Every variable needs a statically-declared data type
- Scope of variables
- Syntax
 - Semicolons at the end of **statements**
 - Braces around blocks of code
 - Keywords

Sept 14, 2011

Sprenkle - CSCI209

55

String method: equalsIgnoreCase

- Does what it's named!

```
String string1 = "Hello";
String string2 = "hello";
boolean test;
test = string1.equalsIgnoreCase(string2);
```

- **test** is true!

Sept 14, 2011

Sprenkle - CSCI209

56

String method: charAt

- A String is a collection of chars

```
String testString1 = "Demonstrate Strings";
char character1;
char character2 = testString1.charAt(3);
character1 = testString1.charAt(2);
```

Sept 14, 2011

Sprenkle - CSCI209

57