

Objectives

- Basics of Java Syntax
- Java fundamentals
 - Primitive data types
 - Arithmetic operators
 - Relational operators

Sep 8, 2008

Sprenkle - CS209

1

Review

- What are qualities of good software?
- What is Java?
- Linux:
 - What is the syntax of the `cp` command?
 - How do you copy an entire directory?
 - How do you make a directory?
 - How do you view the contents of a directory?

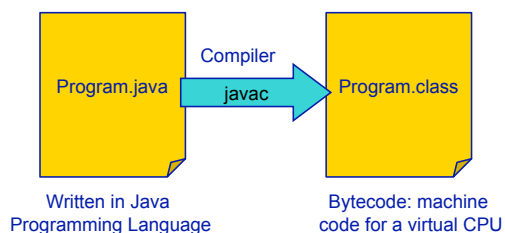
Sep 8, 2008

Sprenkle - CS209

2

Review: Java Programming Language

- Entirely object-oriented
- Similar to Python

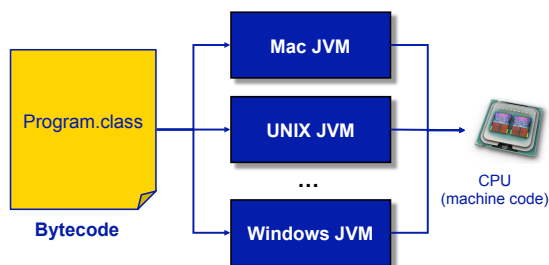


Sep 8, 2008

Sprenkle - CS209

3

Java Virtual Machine (JVM)



- Same **bytecode** executes on each platform

Sep 8, 2008

Sprenkle - CS209

4

Benefits of Java

- Rapid development of programs
 - Large library of classes, including GUIs, Enterprise-level applications, Web applications
- Portability
 - Run program on multiple platforms without recompiling
- Statically-typed language
 - Compiling - find some errors before execution, performance benefits

Sep 8, 2008

Sprenkle - CS209

5

Python Review

```

# a Python program
def main():
    print "Hello"

main()
  
```

- What does this program do?

Sep 8, 2008

Sprenkle - CS209

6

First Java Program

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

Sep 8, 2008

Sprenkle - CS209

7

First Java Program

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

- Everything in Java is inside a **class**
 - Java is *entirely* object-oriented
 - This class is named **Hello**

Sep 8, 2008

Sprenkle - CS209

8

First Java Program

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

Blocks of code marked with { }

Defines the class "Hello"

- In general, each Java program file contains **one** class definition
 - Will have exceptions
- Name of the class is name of file
 - E.g., `Hello.java`

Sep 8, 2008

Sprenkle - CS209

9

First Java Program

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

Access Modifier:
controls if other classes can use code in this class

Sep 8, 2008

Sprenkle - CS209

10

First Java Program

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

method

- Class contains one method: **main**

Sep 8, 2008

Sprenkle - CS209

11

First Java Program: main Methods

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

- Similar to `main` in Python
 - But must be associated with a **class**
- Must take one parameter: an *array* of Strings
- Must be **public static**
- Must be **void**: data type of what method returns (nothing)
- main** is automatically called when program is executed from command line

Sep 8, 2008

Sprenkle - CS209

12

First Java Program

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

- Method contains one line of code
 - What do you think it does?

Sep 8, 2008

Sprenkle - CS209

13

First Java Program: Print Statements

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

- Calls the `println` method on the `System.out` object
- `println` takes one parameter, a `String`
- Displays string on terminal, terminates the line with new line (`\n`) character

Sep 8, 2008

Sprenkle - CS209

14

First Java Program: Comments

```
/**
 * Our first Java class
 * @author Sara Sprenkle
 */
public class Hello {
    public static void main(String[] args) {
        //print a message
        System.out.println("Hello");
    }
}
```

- Comments: `/* */` or `//`

Sep 8, 2008

Sprenkle - CS209

15

Code Style

```
/**
 * Our first Java class
 * @author Sara Sprenkle
 */
```

- **Comments** at top of program
 - Must include your name
 - High-level description of program
- Proper **indentation**
 - Similar to Python
 - Everything within sets of `{}` is indented the same

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

Sep 8,

16

Where are the Differences?

```
# a Python program
def main():
    print "Hello"

main()
```

```
/**
 * Our first Java class
 * @author Sara Sprenkle
 */
public class Hello {
    public static void main(String[] args) {
        //print a message
        System.out.println("Hello");
    }
}
```

7

Java vs Python

- **Semantics** the same, **syntax** different
 - Blocks of code
 - End statements
- Access modifiers
- Data type declarations
- Class-based programs
- Compiled
- We'll see more differences as we go...

Sep 8, 2008

Sprenkle - CS209

18

Translate to Python Program

```
/**
 * Our first Java class
 * @author Sara Sprenkle
 */
public class Hello {
    public static void main(String[] args) {
        //print a message
        System.out.println("Hello");
    }
}
```

Sep 8, 2008

Sprenkle - CS209

19

Translation to Python Program

```
class Hello:
    """Our first Python class"""

    def __init__(self):
        # fill in later...

    def main(self):
        print "Hello"
```

Semi-literal translation

Sep 8, 2008

Sprenkle - CS209

20

JAVA FUNDAMENTALS

Sep 8, 2008

Sprenkle - CS209

21

Print Statement

- Syntax:

```
System.out.println(<String>);
System.out.print(<String>);
```

No newline

- Similar to Python's `file.write()` method
 - Need to combine parameter into one String using `+`'s
 - Python's print used *commas*
 - More on String operations later

Sep 8, 2008

Sprenkle - CS209

22

Escape Sequences

- Same as Python
 - Newline character (carriage return) -> `\n`
 - Tab -> `\t`
 - Quote -> `\"`
 - Backslash -> `\\`
- Note that in Java, you can print a `'` without escaping
- What does the following display?


```
System.out.println("To print a \\", you must use \"\\\\\\\\\\\\\\\\");
```

Sep 8, 2008

Sprenkle - CS209

First.java 23

Java keywords/reserved words

- Case-sensitive
- Can't be used for variable or class names
- Seen so far ...
 - `public`
 - `class`
 - `static`
 - `void`
- Exhaustive list
 - http://java.sun.com/docs/books/tutorial/java/nutsandbolts/_keywords.html

Sep 8, 2008

Sprenkle - CS209

24

Data Types

- Java is **strongly-typed**
 - Every variable must be a **declared type**
- All data in Java is an **object** except for the **primitive data types**:

int	4 bytes (-2,147,483,648 -> 2,147,483,647)
short	2 bytes (-32,768 -> 32,767)
long	8 bytes (really big integers)
byte	1 byte (-128 -> 127)
float	4 bytes (floating point)
double	8 bytes (floating point)
char	2 bytes (Unicode representation), single quotes
boolean	false or true

Variables

- Must be **declared** before used
 - Syntax:** `<datatype> <name> [= value];`
 - Optional assignment
- Variable names typically start with lowercase letter
 - '_' also a valid first character
 - Convention: Subsequent words are capitalized
 - Called "Camel Casing"

Sep 8, 2008

Sprenkle - CS209

26

Variable Examples

- Must be **declared** before used
 - Syntax:** `<datatype> <name> [= value];`
- Examples:
 - `int x;`
 - `double pi = 3.14;`
 - `char q = 'p';`
 - `boolean isValid = false;`

Note must use single quotes for chars

Camel Casing

Sep 8, 2008

Sprenkle - CS209

27

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

Sep 8, 2008

Sprenkle - CS209

28

More Data Type Information

- Default data types
 - Same as Python
 - Result of integer division is an `int`
 - Example: `1/2 = ??`
- Casting
 - Same as Python for primitive types
 - Example: `1/(double) 2`

Sep 8, 2008

Sprenkle - CS209

29

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?

Sep 8, 2008

Sprenkle - CS209

30

Benefits of Static Typing

- Easier to remember type of variable
 - Know what operations that can be executed on a variable
- More benefits later this semester

Sep 8, 2008

Sprenkle - CS209

31

More Why Java?

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

Sep 8, 2008

Sprenkle - CS209

32

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;
```

Sep 8, 2008

Sprenkle - CS209

33

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 later)

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

Sep 8, 2008

Sprenkle - CS209

34

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: ++, --

Conversion.java

Sep 8, 2008

Sprenkle - CS209

35

Output Redirection: >

- In UNIX, we can redirect output to a file
 - For example,
 - ls *.java > java_files.out
 - Above command saves the output from the **ls** command into the file named **java_files.out**
- This is how you will save output from your Java programs initially
 - For example
 - java Intro > out

Sep 8, 2008

Sprenkle - CS209

36

Programming Assignment 0

- Write a program called Intro.java
 - Displays information about yourself
- Fix compiler and logic errors in a program
- Write a simple program using arithmetic
- See Course Web Page
 - [Schedule page](#)
 - [Bookmark it!](#)