

Objectives

- More Java fundamentals
 - Static data types
 - Arithmetic operators
 - Relational operators
 - `java.lang` classes: `Math` and `String` class

Aug 28, 2020

Sprenkle - CSCI209

1

1

Review

- Why Java?
- How do you compile and run Java programs?
- How do you display output in Java?
- What are the modifiers for the `main` method?
 - How do you *call* the `main` method?
- How does Java compare to Python (so far)?

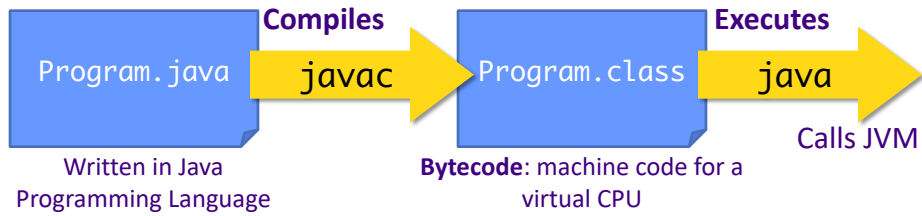
Aug 28, 2020

Sprenkle - CSCI209

2

2

Review: Compiling, Executing Java Programs



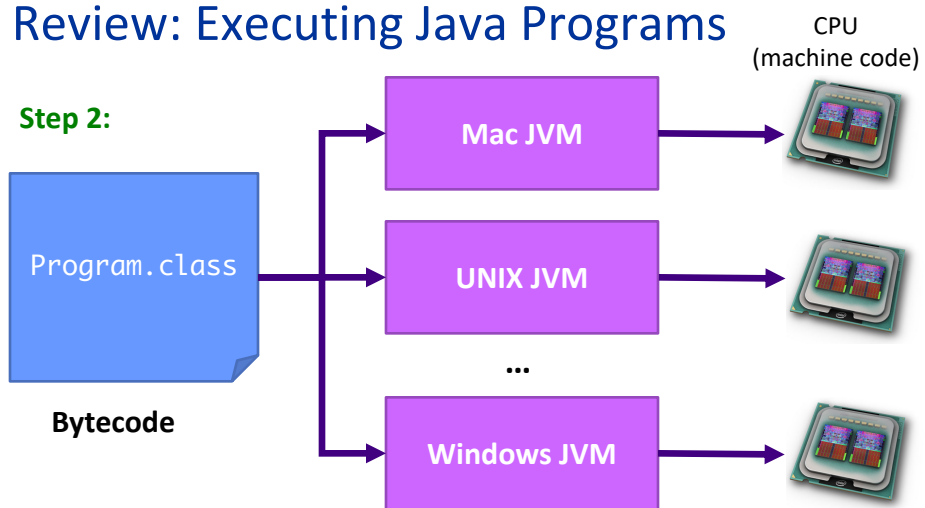
Aug 28, 2020

Sprenkle - CSCI209

3

3

Review: Executing Java Programs



- Same **bytecode** is executed on each platform
- Don't need to provide the source code

Aug 28, 2020

Sprenkle - CSCI209

4

4

Review: Example Java Program

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

Aug 28, 2020

Sprenkle - CSCI209

5

5

Review: Assign 0

- How did it go?
 - How long did it take?
 - What tips/tricks did you learn/would you recommend?
- Unix commands:
 - How do you make a directory?
 - How do you view the contents of a directory?
 - How do you go into a directory?

Aug 28, 2020

Sprenkle - CSCI209

6

6

Commenting

- I am “overcommenting” the Java programs
 - Explaining things that you don’t need to explain
- Expectations for your assignments
 - High-level comment at top of program
 - Explain blocks of code or difficult code
 - Comments that help you
 - Expectations will keep being refined as we learned more

Aug 28, 2020

Sprenkle - CSCI209

7

7

MORE JAVA FUNDAMENTALS

Aug 28, 2020

Sprenkle - CSCI209

8

8

Java keywords/reserved words

- Case-sensitive
- Can't be used for variable or class names
- Reserved words seen so far ...
 - **public**
 - **class**
 - **static**
 - **void**
- Exhaustive list
 - http://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html

Aug 28, 2020

Sprenkle - CSCI209

9

9

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

Aug 28, 2020

Sprenkle - CSCI209

10

10

Variables

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

Aug 28, 2020

Sprenkle - CSCI209

11

11

Variable Examples

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

Note must use single quotes for chars

↑
Camel Casing



Aug 28, 2020

Sprenkle - CSCI209

12

12

Python Transition **Warning**

You can**not** 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

Aug 28, 2020

Sprenkle - CSCI209

13

13

More Data Type-Related Information

- Result of integer division is an **int**

- Same as Python 2, **not** Python 3
- Example: $4/3 = ??$

- Casting

- Similar to Python for primitive types
- Example: $4/(\text{double}) 3$

TestScore.java

Aug 28, 2020

Sprenkle - CSCI209

14

14

Floats in Java

- Decimal literals are considered *doubles*
- This code won't compile:

```
float f = 3.14;
```

Compiler reads 3.14
as a *double*

- Compiler error message:

```
Float.java:15: error: incompatible types: possible
lossy conversion from double to float
    float f = 3.14;
               ^
1 error
```

- To fix code, add an **f** to specification of number or **declare as double**

Aug 28, 2020

Sprenkle - CSCI209

Float.java

15

15

Escape Sequences

- Same as Python:

Meaning	Sequence
Newline character (carriage return)	\n
Tab	\t
Quote	\"
Backslash	\\

- In Java, you can print a ' without escaping
- What does the following display?

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

EscapeCharacters.java

Aug 28, 2020

Sprenkle - CSCI209

16

16

Formatted Output

- `printf` or `format`
 - `System.out` is a `PrintStream` object

```
double d1=3.14159, d2=1.45, total=9.43;

// simple formatting...
System.out.printf("%6.5f and %5.2f ", d1, d2);

// %n is platform-specific line separator,
// e.g., \n or \r\n
System.out.printf("%-6s%5.2f%n", "Tax:", total);
```

Aug 28, 2020

Sprenkle - CSCI209

Format.java

17

17

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?

Aug 28, 2020

Sprenkle - CSCI209

18

18

Class Constants

- Constant variable for all methods in class or for multiple classes
- Requires **static** keyword
 - **static**: “for class”
 - Also used for methods (will see more later)

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

Aug 28, 2020

Sprenkle - CSCI209

19

19

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: ++, --

Aug 28, 2020

Sprenkle - CSCI209

Conversion.java 20

20

INTRODUCTION TO JAVA LIBRARIES

Aug 28, 2020

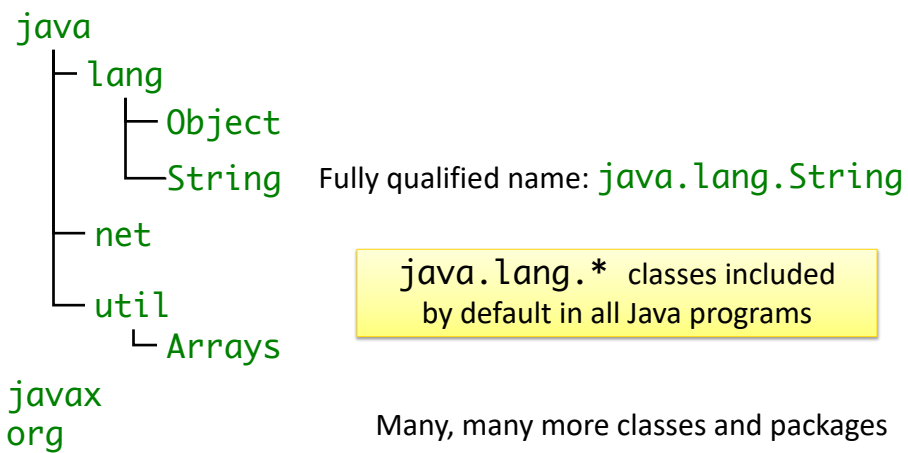
Sprenkle - CSCI209

21

21

Java Libraries

- Organized into a hierarchy of *packages*



Aug 28, 2020

Sprenkle - CSCI209

22

22

Java API Documentation

<https://docs.oracle.com/en/java/javase/14/docs/api/index.html>

- **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
- Bookmark it!
 - Too many classes, methods to remember them all
 - Refer to it often

Aug 28, 2020

Sprenkle - CSCI209

23

23

`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
 - <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/Math.html>
 - Note how API is specified

Aug 28, 2020

Sprenkle - CSCI209

24

24

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;
  
```

method

constant

Use `Classname.methodname()`
to call Math's methods because they're **static**
static: for (or of) the class

MathExample.java

Aug 28, 2020

Sprenkle - CSCI209

25

25

java.lang.String class

- Similar functionality to Python use differently
 - Mostly *methods*!
- Strings are represented by **double** quotes: ""
 - Single quotes represent **chars** only
- Examples:

```

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

Aug 28, 2020

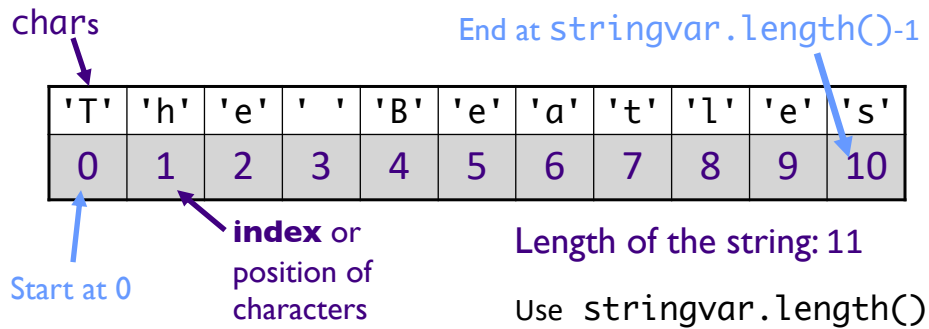
Sprenkle - CSCI209

26

26

Strings

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



Use `charAt` method to access chars

Aug 28, 2020

Sprenkle - CSCI209

27

27

String method: `charAt`

- A `String` is a collection of chars

```
String testString1 = "Demonstrate Strings";
char character1;
char character2 = testString1.charAt(3);
character1 = testString1.charAt(testString1.length()-2);
System.out.println(character1 + " " + character2);
```

Displays "g o"

Aug 28, 2020

Sprenkle - CSCI209

28

28

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-1`

```
String language = "Java!";
String subStr = language.substring(1);
String subStr2 = language.substring(2, 4);
```

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

29

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"

Aug 28, 2020

Sprenkle - CSCI209

30

30

Review: 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;
double testScore = (double) earnedPoints/totalPoints;

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

↑
Converted to a `String`

Aug 28, 2020

Sprenkle - CSCI209

31

31

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

Aug 28, 2020

Sprenkle - CSCI209

32

32

String methods: and many more!

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

<https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/String.html>

Aug 28, 2020

Sprenkle - CSCI209

33

33

To Do

- Assign 1
 - Part 0: Fixing compiler and logic errors from program
 - Part 1: Finding the file extension of a filename
 - Due Monday before class

Aug 28, 2020

Sprenkle - CSCI209

34

34