

Lesson 5

(5 min)

Logistics

- Join Remind Thread — link in email
- Make sure to have completed the Java course on Codecademy
- Any ideas for topics for guest speakers?

(5 min)

Cool Alexa Skill demo: 1-2 students

Talk about it with respect to 7 UX factors

(15 min)

Java intro:

- What are classes?
 - A blueprint / template for an object
 - Defines the objects properties and behaviors
- Instance variables?
 - These are the object's properties / attributes / characteristics
 - Protected vs. Private vs. Public
- Methods?
 - These are the object's behaviors / capabilities
- Constructor?
 - Special type of method used to initialize an instance of an object
 - Invoked at the time of object creation
 - Look like method declarations, except they use the name of the class and have no return type
 - Object can have multiple constructors, to represent different ways in which you can initialize the object
 - 2 types:
 - Default (no arguments)
 - Parametrized (has arguments)
 - If you don't provide a constructor in the class, the compiler automatically creates a default constructor
- Inheritance?
 - The process where one class acquires the properties (methods and fields) of another
 - Makes information more manageable in a hierarchical order
 - Class which inherits the properties of other is known as subclass (child class)
 - Class whose properties are inherited is known as superclass (base class, parent class)
- Abstract classes?
 - Contain one or more abstract methods

- a method that is declared but contains no implementation
- cannot be instantiated
- requires subclasses to provide implementations for the abstract methods
- Example: Animal <—> Dog, Cow
 - both makeNoise but in a different way (good candidate for abstract method)
 - both sleep(), in the same way, could be implemented in abstract class and inherited

Java demo:

- abstract Shape class
- Rectangle subclass
- Demo for today main method
- Comments — header, in line

(15 min)

Breakout Groups — Zoom

- Write your own Shape subclass (in a Google doc):
 - Circle
 - Triangle
 - Ellipse
 - Hexagon (Regular)
- Implement the two required methods: calculateArea() and calculatePerimeter()
- Implement a third method of your choice

(15 min)

Whole class — Test the subclasses

- Copy / paste code from Google docs into Eclipse
- Run test code from main
- Debugging techniques
- If time, Java docs: <https://docs.oracle.com/javase/7/docs/api/overview-summary.html>
- More useful to search for a specific class, like Java String:
<https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>

(5 min)

Wrap up:

- Assignments for the weekend — might take longer
- Lab 1 — GitHub
- Lab 2 — Eclipse / local dev environment
- Google if run into issues
- Reach out to me if you're struggling with set up