

Lecture 23:

Resizing, Iterators, Abstract Classes, Wrap up!

08/17/22



Upcoming

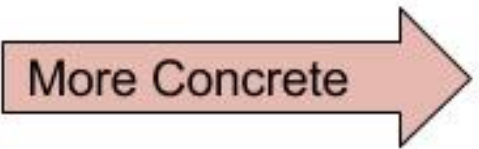
- 20 Questions resub due Saturday 8/20 @11:59pm
- Last day of IPL is today!

- Final Exam
 - Thursday 8/18: Final Exam part 1 in your section
 - Friday 8/19: Final Exam part 2, 10:50 - 11:50am

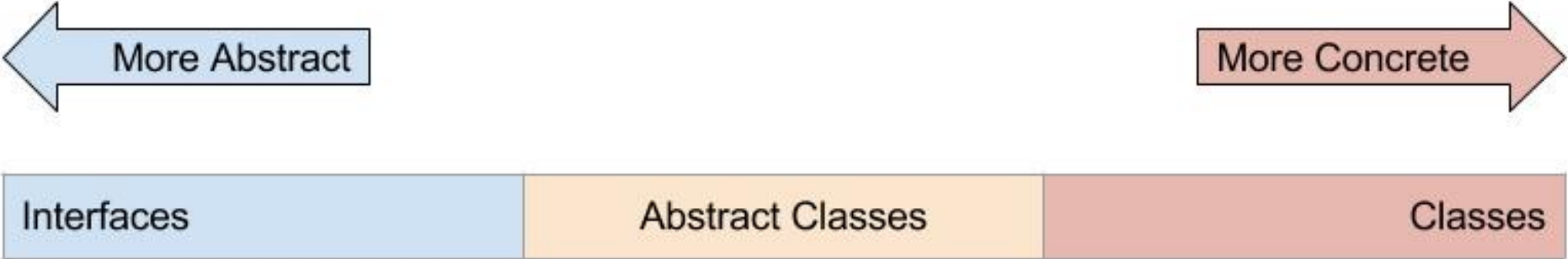
- Exam reminders
 - Bring your Husky ID
 - Show up early
 - Sleep!

Misc

- Please fill out the evaluations! Close Friday 8/19 @ 11:59pm
 - Course eval – 1 point extra credit!
 - TA eval – please do it 😊
- Nominate your TA for a Bob Bandes Award! (see Ed)



Abstract Classes

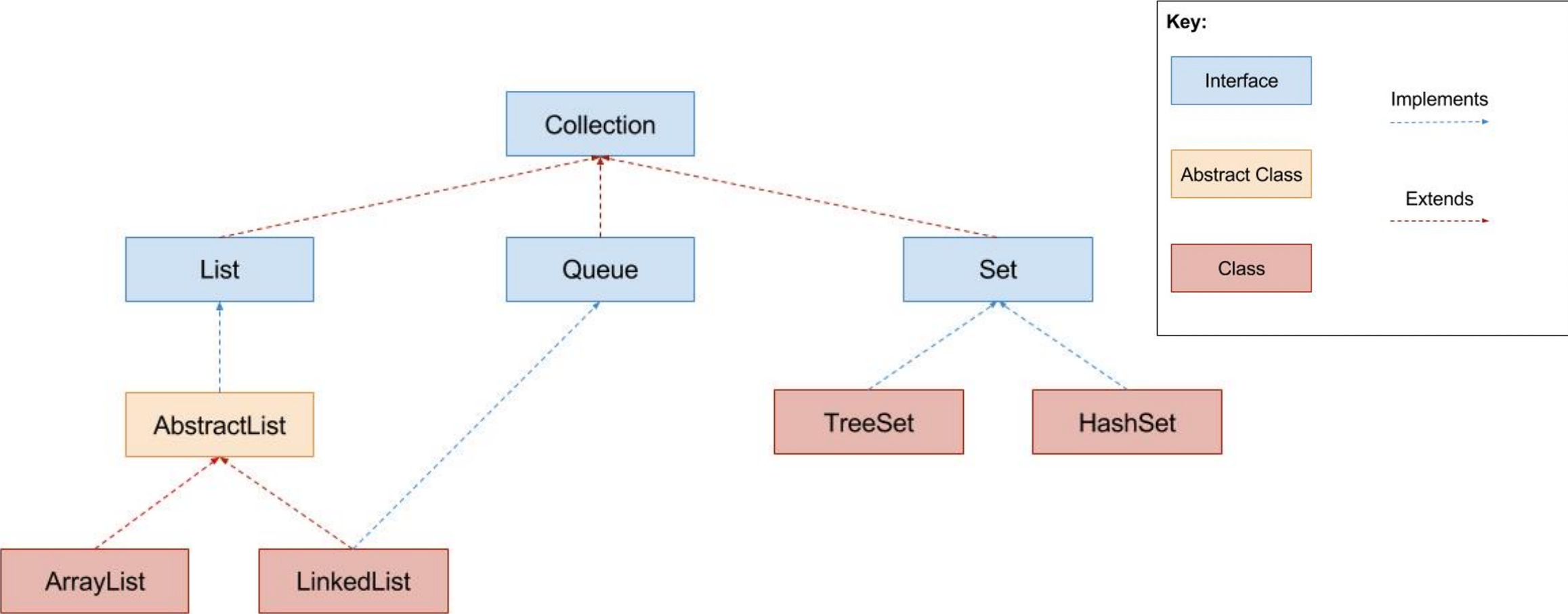


```
public abstract class Animal {
    /*
    * Can't do this since this class is abstract
    * Animal a = new Animal();
    */

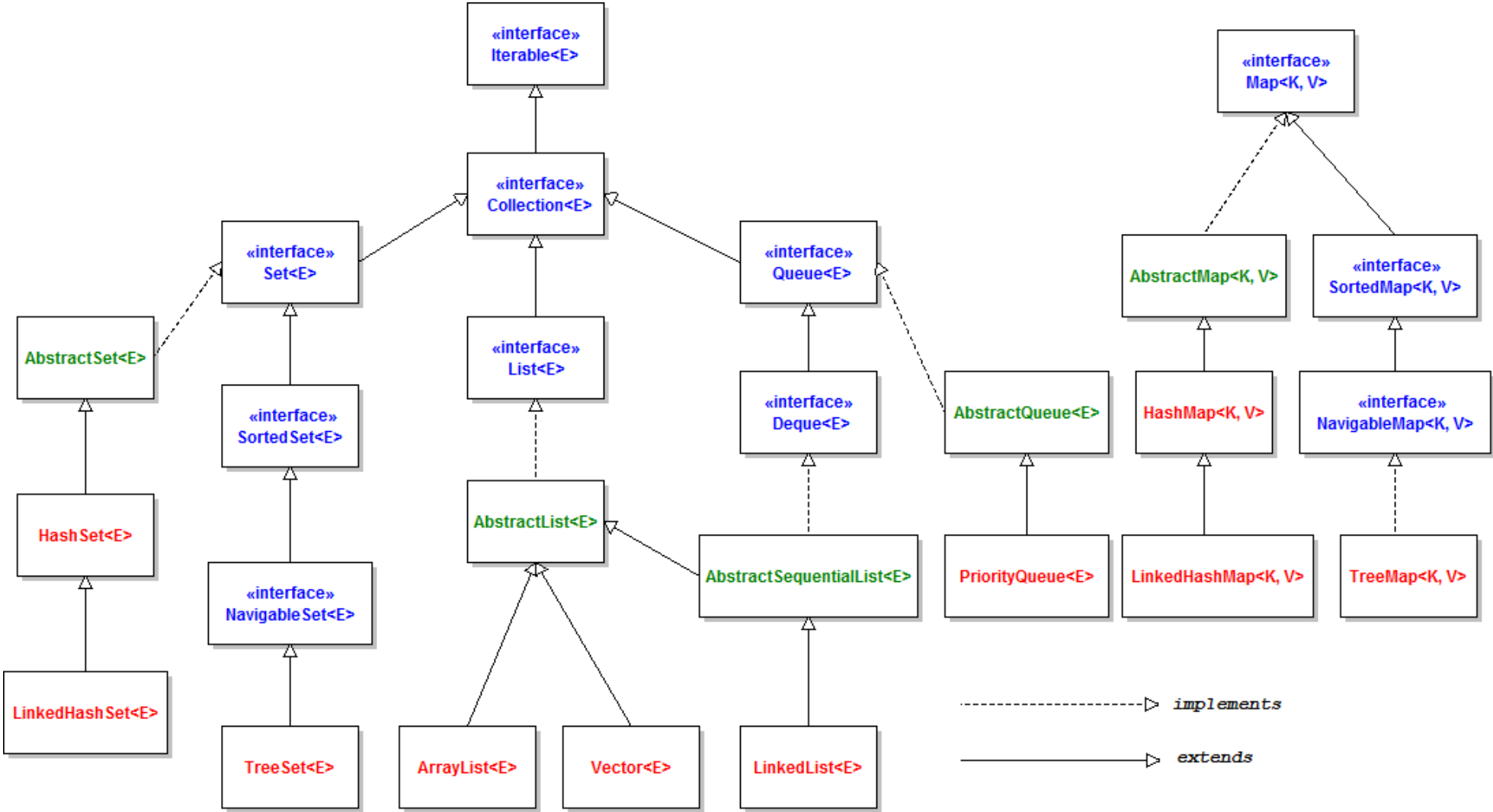
    public abstract void animalSound();

    public void sleep() {
        System.out.println("Zzz");
    }
}
```

Java Collections Hierarchy (simplified)



Java Collections Hierarchy (full version)



We did a lot this quarter!

CS Concepts

- Client/Implementer
- Efficiency
- Recursion
- Grammars
- Searching / Sorting
- Backtracking
- Hashing
- Huffman Compression

Data Structures

- Lists
- Stacks
- Queues
- Sets
- Maps
- Priority Queues

Java Language

- Exceptions
- Interfaces
- References
- Comparable
- Generics
- Inheritance / Polymorphism

Java Collections

- Arrays
- ArrayList
- LinkedList
- Stack
- TreeSet / TreeMap
- HashSet / HashMap
- PriorityQueue

What project?

- Add a GUI to the random sentence generator
- Automate chemistry, physics, calculus problems, etc
 - Maybe even automate writing code with good style?
- Find quotes by keyword in books
- What are you currently doing that a computer could do?
- [List of some project ideas](#)

What language?

- Expanding your Java knowledge with a project is valuable
- Pick a project, see what similar projects use!
 - iOS: [Swift](#)
 - Android: Java, Kotlin
 - Client-side web: [Javascript](#) (many frameworks to choose from)
 - Beautiful visuals: [Processing](#)
 - Data Processing + Machine Learning: [Python](#)
 - Data Management: [SQL](#)
 - Embedded systems: C / C++
- Learn a new programming paradigm
 - Functional languages: [Racket](#), [Haskell](#), [Scala](#), (now, Java 8!)

Leveraging existing code

Here are just a FEW examples. There is so much more!

- Processing language
 - <http://nlp.stanford.edu/software/>
- Building games
 - <http://lwjgl.org/>
 - <http://jbox2d.org/> (with physics!)
- Processing biological data
 - http://biojava.org/wiki/Main_Page
- Accessing Facebook data
 - <http://restfb.com/>
- Making music
 - <http://www.jfugue.org/>

Courses?

- CSE non-majors

- CSE 154: Web Programming
- CSE 163: Intermediate Data Programming (Python)
- CSE 373: Data Structures and Algorithms
- CSE 374: Programming Concepts and Tools (C/C++, Linux, ...)
- CSE/STAT 416: Machine learning (requires STAT 311 or 390)
- CSE 131: Digital Photography
- CSE 460: Animation Capstone (open to all majors)
- And more!

- CSE majors

- CSE 311: (Mathematical) Foundations of Computing
- CSE 332: Data Abstractions (Data Structures and Algorithms)
- CSE 331: Software Design and Implementation
- CSE 341: Programming Languages
- CSE 344: Intro to Data Management (and databases)
- CSE 351: Hardware/Software Interface
- And more!

- INFO, AMATH, HCDE, DXARTS, ...

Beyond programming

- [Investigate how to best distribute relief funds](#)
- [Digitize basketball players](#)
- [Help deaf/hard-of-hearing people identify sounds](#)
- [Detect and prevent toxicity online](#)
- [Recognize disinformation online](#)
- [Make movies](#)
- [Improve digital collaboration](#)
- [Design algorithms that are more fair and better respect privacy](#)
- [Fix Olympic badminton](#)
- And so much more!

Almost done!!

