We assume that you are proficient at object-oriented programming in Java.

Instructors
- **Lec 1 & 2**: Deb Deppeler, 5376 CS, deppeler@cs.wisc.edu
- **Lec 3**: Alexi Brooks, 4397 CS, albrooks@cs.wisc.edu

Course Website: [https://canvas.wisc.edu/](https://canvas.wisc.edu/)
If Canvas is down, we may post updates on: [http://pages.cs.wisc.edu/~cs367-1/](http://pages.cs.wisc.edu/~cs367-1/)

See modules links for online readings and lecture outlines (no textbook)
**Waitlisted?** Continue attending. Some seats might open.

**Homework 0 available**

**Today**
- Collections
  - bag intro
  - abstract data types and data structures
  - designing an Integer Bag ADT – Java interfaces
  - using the Integer Bag ADT – review of autoboxing
- Characteristics of Good & Reusable Software
- Generalizing the Integer Bag ADT – Java Object
- Implementing a General Bag ADT

**Next Time**
- **Read**: *Introduction*, start *Lists*
- Implementing the Bag ADT
  - casting when using Object
  - using Java generics for generality
- List ADT
  - designing the ListADT
  - coding the ListADT as a Java interface
Collections

→ What is a collection?

→ What operations can you do on a collection? Which are the most fundamental?
Example: Bags

Concept

Operations

Problems

→ What problems might occur when doing Bag operations?
What is Abstract?

What is Data? Type?
ADTs vs. Data Structures

Abstract Data Type (ADT)

Data Structure (DS)
Designing an Integer Bag ADT

Conceptual Description

Public Interface

Coding Issues
Example 1: Using the Integer Bag ADT

→ Write a code fragment
to put the numbers 0 through 99 into an Integer Bag ADT named bag.

    IntegerBagADT bag = new ...;  //assume bag has been instantiated here

Java Autoboxing Review
Example 2: Using the Integer Bag ADT

Complete the printBag method
so that it prints the contents of the parameter bag.

Challenge: Implement your printBag method so that it doesn’t change the bag’s contents.

public static void printBag(IntegerBagADT bag) {

What makes software good?

What makes code reusable?
Generalizing the Integer Bag ADT

→ What class in Java can be used to reference any Java object?

→ What modifications are needed to generalize our Integer Bag ADT?

import java.util.*;

public interface IntegerBagADT {

    void add(Integer item);
    
    Integer remove() throws NoSuchElementException;

    boolean isEmpty();

}
Implementing BagADT Using an Array of Object References

public class ArrayBag

    //instance variables

    //constructor

    //BagADT methods

    // could add other methods specific to the array implementation
}