CS 367 - Introduction to Data Structures  
Monday, June 15, 2015  

Course website: http://pages.cs.wisc.edu/~cstapleton/367/  
Piazza: https://piazza.com/wisc/summer2015/cs367/  

Instructor  
Cea Stapleton  
cstapleton@cs.wisc.edu ; Use Piazza for course related questions.  

TA  
Haseeb Tariq  
haseeb@cs.wisc.edu ; Use Piazza for course related questions.  

Today  
- Course Introduction & Logistics -- see Syllabus  
- Sign up for Piazza!  
- Characteristics of Good & Reusable Software  
- Collections  
- Generics & Abstraction  
- Interfaces  
- Abstract Data Types  
- Sack ADT: Examples & Implementation  
- HW0 Assigned  

Next Time  
- Read: Introduction & Lists @ http://pages.cs.wisc.edu/~cstapleton/367/calendar/  
- Begin: HM0 (due Wednesday!)  
- Review SackADT  
- Generics & the ListADT  
- ListADT Concept & Design  
- ListADT Use  
- ListADT Implementation
Course Introduction & Logistics -- see Syllabus

* We assume that you are proficient at object-oriented programming in Java. If you have not learned object-oriented programming, you should complete CS 302 first. If you have learned object-oriented programming in a language like C++, you should focus time in the next two weeks to learn Java or consider taking CS 302.

Sign up for Piazza!

https://piazza.com/wisc/summer2015/cs367/

Characteristics of Good & Reusable Software

What makes software good?

What makes code reusable?

Collections

What is a collection?

What operations can you do on a collection? Which are the most fundamental?
Generics & Abstraction

What is a generic?

What is generic programming?

Interfaces

What is an Interface?

Why would we want to use one?
Abstract Data Types

What is an ADT?

Why would we want to use an ADT?

What makes an ADT?
Sack ADT: Examples & Implementation

Conceptual Picture

Public interface/operations

```java
import java.util.*;

public interface SackADT
{
}
```

Implementation Challenges

Write a code fragment to put the numbers 0 through 99 into the sack:
```java
Sack s = new Sack();
```

Java Autoboxing
The Sack ADT and Java objects

```java
import java.util.*;

public interface SackADT
{
    void add(Object item);
    Object remove() throws NoSuchElementException;
    boolean isEmpty();
}
```

Why are we using the Object class in our SackADT interface?
Implementing SackADT using an Array of Object References

import java.util.*;

public class ArraySack
{

}