

# Announcements

- Program 3, late due date: 10/15 at 11:59PM, if you submit for the late due date it minus 5% on your program grade.
- Zybooks assignment dates have changed
  - The zybooks chapters on my schedule were inaccurate
- Schedule has been adjusted
- Program 4 should be released sometime this week, it's due date will match when it is released.

# Generics

- A **generic type** is a generic class or interface that is parameterized over types.
  - In other words, you use a generic type when you do not know what type you will be working with.
- A **generic class** is any class that uses a generic type.
- The syntax for using a generic type is: `<E>` applied next to the class name.  
For example:
  - `public class Foo<E> { ...`
  - The above states that class Foo will be using a generic type called E

# Generics (cont.)

The choice of the letter: E as a generic type is based on the following naming convention:

**Generic type parameter** names should be single, uppercase letters.

- E - element
- K - key
- N - number
- T - type
- V - value
- S, U . etc. - 2nd, and 3rd generic types

# Generics (Cont.)

- A **generic method** is a method definition having a generic type parameter that may be used in place of types in the method
- The generic type is placed in the signature of the method, before the return type, like so:
  - `public <E> void recieveType(E obj) {...`

Let's look at an example...

