Generic Methods and Classes

Weiss, sections 4.2 through 4.5
(pages 125–141)

I am indebted to my colleague, John David Stone, for assistance with this handout. This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit https://creativecommons.org/licenses/by-sa/4.0/ or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.
Generic Classes

- Generic classes are declared using type parameters enclosed in <>
- Type parameters have to be reference types.

```java
public class GenericMemoryCell<AnyType>
{
    public AnyType read()
    {
        return storedValue;
    }
    public void write( AnyType x )
    {
        storedValue = x;
    }

    private AnyType storedValue;
}
```
Wildcards with bounds

```java
public static double totalArea( ArrayList<Shape> arr )
{
    double total = 0;
    for( Shape s : arr )
    {
        if( s != null )
            total += s.area();
    }
    return total;
}

public static double totalArea( ArrayList<? extends Shape> arr )
{
    double total = 0;
    for( Shape s : arr )
    {
        if( s != null )
            total += s.area();
    }
    return total;
}
```
public static <AnyType> AnyType findMax( AnyType [ ] a )
{
    int maxIndex = 0;
    for( int i = 1; i < a.length; i++ )
        if( a[ i ].compareTo( a[ maxIndex ] ) > 0 )
            maxIndex = i;
    return a[ maxIndex ];
}
Type Erasure

• Backward compatibility
• Compiler generates a non-generic (raw) class.
  – Same name as the generic class without type parameters