

# Chapter 4 Notes Inheritance Polymorphism

Student  
Employee

Name  
Number  
Address

Similarity leads to duplicity of code.

Add a parent object



Person



Public Class Derived extends Base

Student Employee



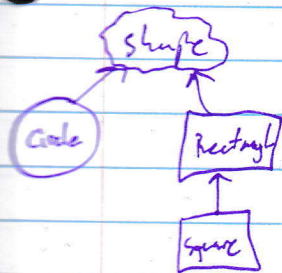
- Any subclass can be treated like a parent

Integrated grad faculty staff - Any child can override a parent method

final methods can't change!

Person [] = { Student, Employee, faculty }

↑ type compatible



Where should area go?

Shape

- abstract area();

Not defined, but children must

Rectangle

- area () { };

Square size, size

- area() { super (size, size) }

Class implements Comparable

public int compareTo (Object)

## Chapter 4 Notes: Generics

### With objects

~~Read~~ Object read() ← must down cast      Java 5+  
write (Object)      Autobox      AutoUnbox  
Object values      write(new Integer)      int a = read

### Adapter classes

Adapter classes are used as a way for 2 classes to be used together  
class StorageCell

write { }

delete { }

Comparable  
↑      ↑  
Integer      String

- type safety is critical

public class GenericMem <Type>

Type read()

return val

void write (Type x)

val = x

- if creation of type has <String> then every Type is replaced with that

Type val;

public interface Compare <Type>

int compareTo (Type)



## 4: Generics pt. 2

totalArea (ArrayList < Shape > )

- generics are not  
co-variant

↳ square won't work

totalArea (ArrayList < ? extends Shape >

↑  
wild card

- still general using  
? extends Base

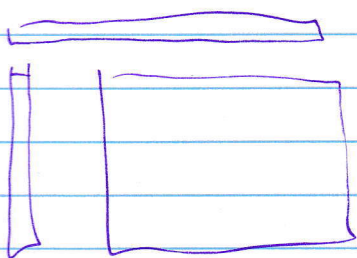
public static <AnyType> boolean contains (AnyType[] x, AnyType y)

public static <AnyType extends Comparable < ? ~~other~~ super AnyType >>

- Primitives can't be used

- instance of only

### The Functor



- which rectangle is larger  
- would be nice to have  
more than one comparator

public interface Comparator <AnyType>

int compare (AnyType lhs, AnyType rhs)

public static <AnyType> AnyType

compareAll (AnyType a, Comparator < ? ~~extends~~ super AnyType >>

class compWid implements Comparator < SimpleRectangle >

4: Generics cont:

### Class Nesting

```
class Comp {  
    private static class OrderByWidth {  
    }  
}
```

- Can be in main
- Can be inline or anonymous

### Dynamic Dispatch

Base

foo (Base x)

foo (Der x)

Der

foo (Base x)

foo (Der x)

- Makes sure to override correctly!
- No multiple inheritance