Comp 401
Inheritance – Inherited Variables and Constructors

Instructor: Prasun Dewan

Prerequisite

- Inheritance

More Inheritance

- Inheritance
  - Graphics Examples
  - Inherited Variables
  - Constructors
  - Memory Representation

Point Interface

```
public interface Point {
   public int getX();
   public int getY();
   public double getAngle();
   public double getRadius();
}
```

Read-only properties defining immutable object!

Class: ACartesianPoint

```
public class ACartesianPoint implements Point {
  int x, y;
  public ACartesianPoint(int theX, int theY) {
    x = theX;
    y = theY;
  }
  public ACartesianPoint(double theRadius, double theAngle) {
     x =  (int) (theRadius*Math.cos(theAngle));
     y = (int) (theRadius*Math.sin(theAngle));
  }
  public int getX() { return x; }
  public int getY() { return y; }
  public double getAngle() { return Math.atan2(y, x); }
  public double getRadius() { return Math.sqrt(x*x + y*y); }
}
```

Extension: Mutable Point

```
public interface MutablePoint extends Point {
  void setX(int newVal);
  void setY(int newVal);
}
```

Calling Constructor in Superclass

```
public class AMutablePoint extends ACartesianPoint
                           implements MutablePoint {
  public AMutablePoint(int theX, int theY) {
    super(theX, theY);
  }
  public void setX(int newVal) {
    x = newVal;
  }
  public void setY(int newVal) {
    y = newVal;
  }
}
```

Superclass constructor

'Inherited" but not available to users of the class if not called from subclass

Bounded Point

```
public interface BoundedPoint extends MutablePoint {
  public MutablePoint getUpperLeftCorner();
  public MutablePoint getLowerRightCorner();
  public void setUpperLeftCorner(MutablePoint newVal);
  public void setLowerRightCorner(MutablePoint newVal);
}
```

Adding Variables

```
public class ABoundedPoint extends AMutablePoint
                                    implements MutablePoint {
  MutablePoint upperLeftCorner, lowerRightCorner;
  public ABoundedPoint(int initX, int initY,
         MutablePoint anUpperLeftCorner,
         MutablePoint aLowerRightCorner) {
    super(initX, initY);
    upperLeftCorner = anUpperLeftCorner;
    lowerRightCorner = aLowerRightCorner;
    fixX();
    fixY();
  }
  void fixX() {
    x = Math.max(x, upperLeftCorner.getX());
    x = Math.min(x, lowerRightCorner.getX());
  }
  void fixY() {
    y = Math.max(y, upperLeftCorner.getY());
    y = Math.min(y, lowerRightCorner.getY());
  }
```

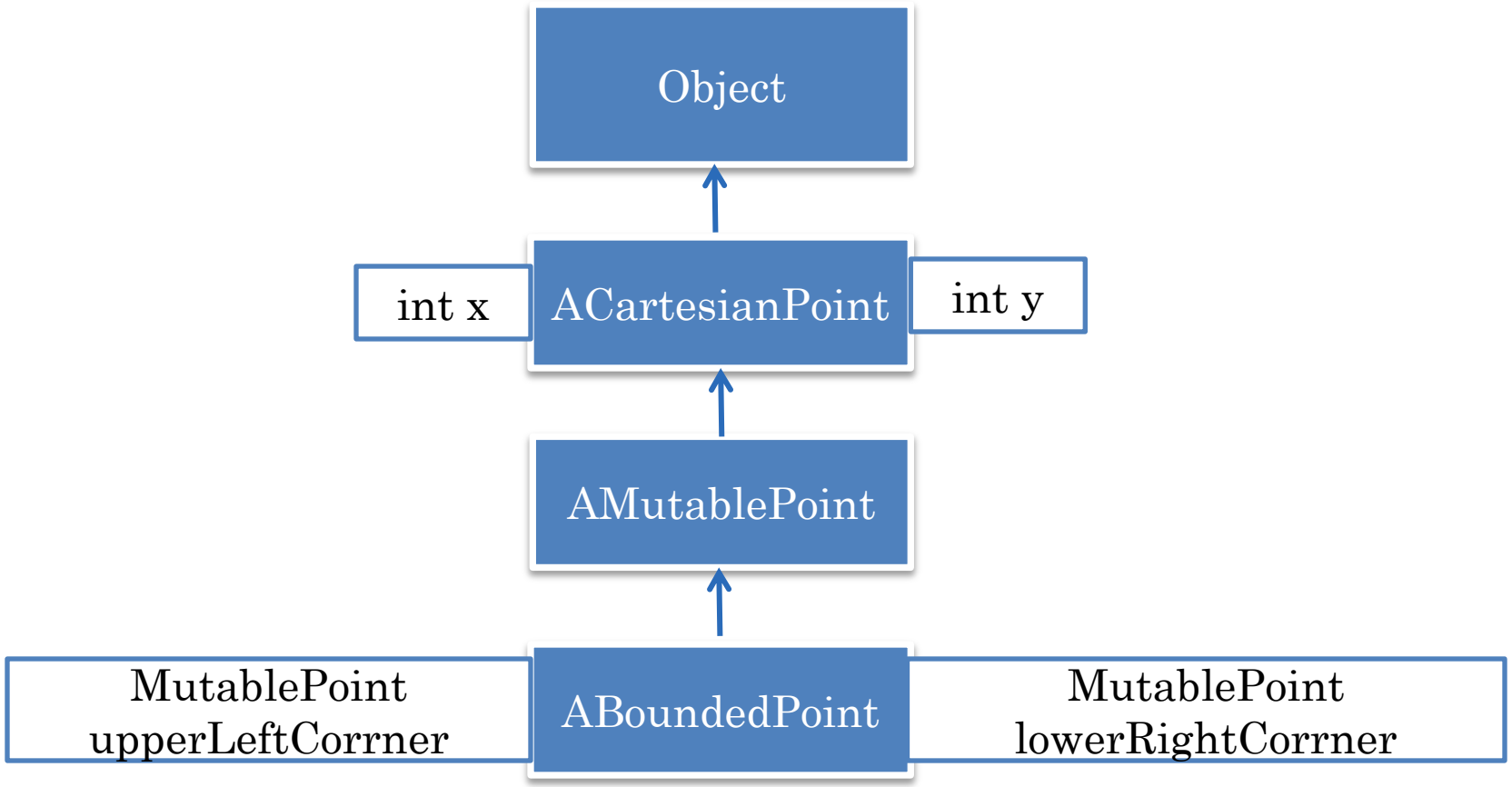
Variables also extended unlike in previous examples

Superclass constructor

Bounded Point Implementation

```
  public void setX(int newVal) {
    super.setX(newVal);
    fixX();
  }
  public void setY(int newVal) {
    super.setY(newVal);
    fixY();
  }
  public MutablePoint getUpperLeftCorner() {
    return upperLeftCorner;
  }
  public MutablePoint getLowerRightCorner() {
    return lowerRightCorner;
  }
  public void setUpperLeftCorner(MutablePoint newVal) {
    upperLeftCorner = newVal;
    fixX();
    fixY();
  }
  public void setLowerRightCorner(MutablePoint newVal) {
    lowerRightCorner = newVal;
    fixX();
    fixY();
  }
}
```

New Class Hierarchy and Variables



Calling Constructor in Superclass

```
public class ABoundedPoint extends AMutablePoint
                                    implements MutablePoint {
  MutablePoint upperLeftCorner, lowerRightCorner;
  public ABoundedPoint(int initX, int initY,
         MutablePoint anUpperLeftCorner,
         MutablePoint aLowerRightCorner) {
    super(initX, initY);
    upperLeftCorner = anUpperLeftCorner;
    lowerRightCorner = aLowerRightCorner;
    fixX();
    fixY();
  }
  void fixX() {
    x = Math.max(x, upperLeftCorner.getX());
    x = Math.min(x, lowerRightCorner.getX());
  }
  void fixY() {
    y = Math.max(y, upperLeftCorner.getY());
    y = Math.min(y, lowerRightCorner.getY());
  }
```

Super call

Calling Constructor in Superclass

```
public class ABoundedPoint extends AMutablePoint
                                    implements MutablePoint {
  MutablePoint upperLeftCorner, lowerRightCorner;
  public ABoundedPoint(int initX, int initY,
         MutablePoint anUpperLeftCorner,
         MutablePoint aLowerRightCorner) {
    upperLeftCorner = anUpperLeftCorner;
    lowerRightCorner = aLowerRightCorner;
    fixX();
    fixY();
    super(initX, initY);
  }
  void fixX() {
    x = Math.max(x, upperLeftCorner.getX());
    x = Math.min(x, lowerRightCorner.getX());
  }
  void
    y
    y
  }
```

Super call

Super call must be first statement in constructor but not other methods.

Subclass may want to override initialization in super class

Superclass vars initialized before subclass vars, which can be used by thelatter

Subclass vars not visible in superclass

Calling Constructor in Superclass

```
public class ABoundedPoint extends AMutablePoint
                                    implements MutablePoint {
  MutablePoint upperLeftCorner, lowerRightCorner;
  public ABoundedPoint(int initX, int initY,
         MutablePoint anUpperLeftCorner,
         MutablePoint aLowerRightCorner)
    upperLeftCorner = anUpperLeftCorner;
    lowerRightCorner = aLowerRightCorner;
    fixX();
    fixY();
  }
  void fixX() {
    x = Math.max(x, upperLeftCorner.getX());
    x = Math.min(x, lowerRightCorner.getX());
  }
  void fixY() {
    y
    y = Math.min(y, lowerRightCorner.getY());
  }
```

No Super call

Java complains that no super call

Wants Super class variables to be initialized

Calling Constructor in Superclass

```
public class ABoundedPoint extends AMutablePoint
                                    implements MutablePoint {
  MutablePoint upperLeftCorner, lowerRightCorner;
  public ABoundedPoint(int initX, int initY,
         MutablePoint anUpperLeftCorner,
         MutablePoint aLowerRightCorner) {
    x = initX;
    y = initY;
    upperLeftCorner = anUpperLeftCorner;
    lowerRightCorner = aLowerRightCorner;
    fixX();
    fixY();
  }
  void fixX() {
    x = Math.max(x, upperLeftCorner.getX());
    x = Math.min(x, lowerRightCorner.getX());
  }
  voi
    y
    y = Math.min(y, lowerRightCorner.getY());
  }
```

Manual initialization

Java complains that no super call

Wants Object variables to be initialized

Missing Super Call

```
public class AStringSet extends AStringDatabase {
   public AStringSet () {
   }
  public void addElement(String element) {
            if (member(element)) return;
            super.addElement(element);
  }
}
```

No complaints

Java Insertion

```
public class AStringSet extends AStringDatabase {
   public AStringSet () {
       super();
   }
  public void addElement(String element) {
            if (member(element)) return;
            super.addElement(element);
  }
}
```

Missing Constructor and Super Call

```
public class AStringSet extends AStringDatabase {
   public void addElement(String element) {
            if (member(element)) return;
            super.addElement(element);
  }
}
```

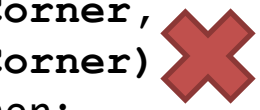
No complaints

Java Insertion

```
public class AStringSet extends AStringDatabase {
   public AStringSet () {
       super();
   }
  public void addElement(String element) {
            if (member(element)) return;
            super.addElement(element);
  }
}
```

Calling Constructor in Superclass

```
public class ABoundedPoint extends AMutablePoint
                                   implements MutablePoint {
  MutablePoint upperLeftCorner, lowerRightCorner;
  public ABoundedPoint(int initX, int initY,
         MutablePoint anUpperLeftCorner,
         MutablePoint aLowerRightCorner)
    upperLeftCorner = anUpperLeftCorner;
    lowerRightCorner = aLowerRightCorner;
    fixX();
    fixY();
  }
  void fixX() {
    x = Math.max(x, upperLeftCorner.getX());
    x = Math.min(x, lowerRightCorner.getX());
  }
  void fixY() {
    y = Math.max(y
    y = Math.min(y, lowerRightCorner.getY());
  }
```



No Super call

Could it insert the super call?

Calling Constructor in Superclass

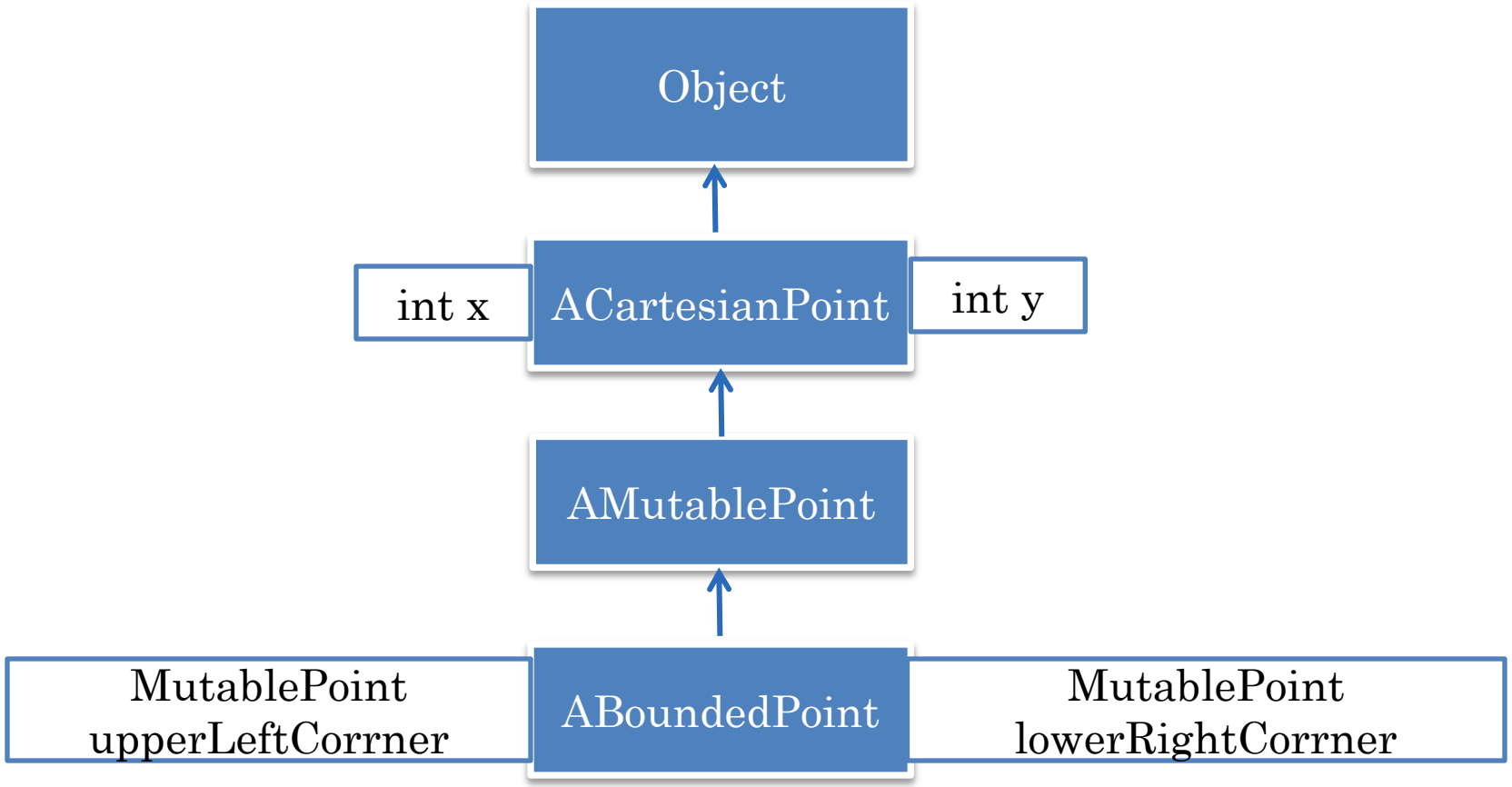
```
public class ABoundedPoint extends AMutablePoint
                                   implements MutablePoint {
  MutablePoint upperLeftCorner, lowerRightCorner;
  public ABoundedPoint(int initX, int initY,
         MutablePoint anUpperLeftCorner,
         MutablePoint aLowerRightCorner) {
    super(initX, initY);
    upperLeftCorner = anUpperLeftCorner;
    lowerRightCorner = aLowerRightCorner;
    fixX();
    fixY();
  }
  void fixX() {
    x = Math.max(x, upperLeftCorner.getX());
    x = Math.min(x, lowerRightCorner.getX());
  }
  void fixY() {
    y = Math.max(y, upperLeftCorner.getY());
    y = Math.min(y, lowerRightCorner.getY());
  }
```

Does not know what the actual parameters should be

Could initialized them to 0s and null values

There may be multiple constructors, which should it choose?

Memory?


Object
int x
ACartesianPoint
int y
AMutablePoint
MutablePoint
upperLeftCorrner
ABoundedPoint
MutablePoint
lowerRightCorrner

Inheritance and Memory Representation

```
public class ACartesianPoint
implements Point {

        int x, y;

    …

}
```

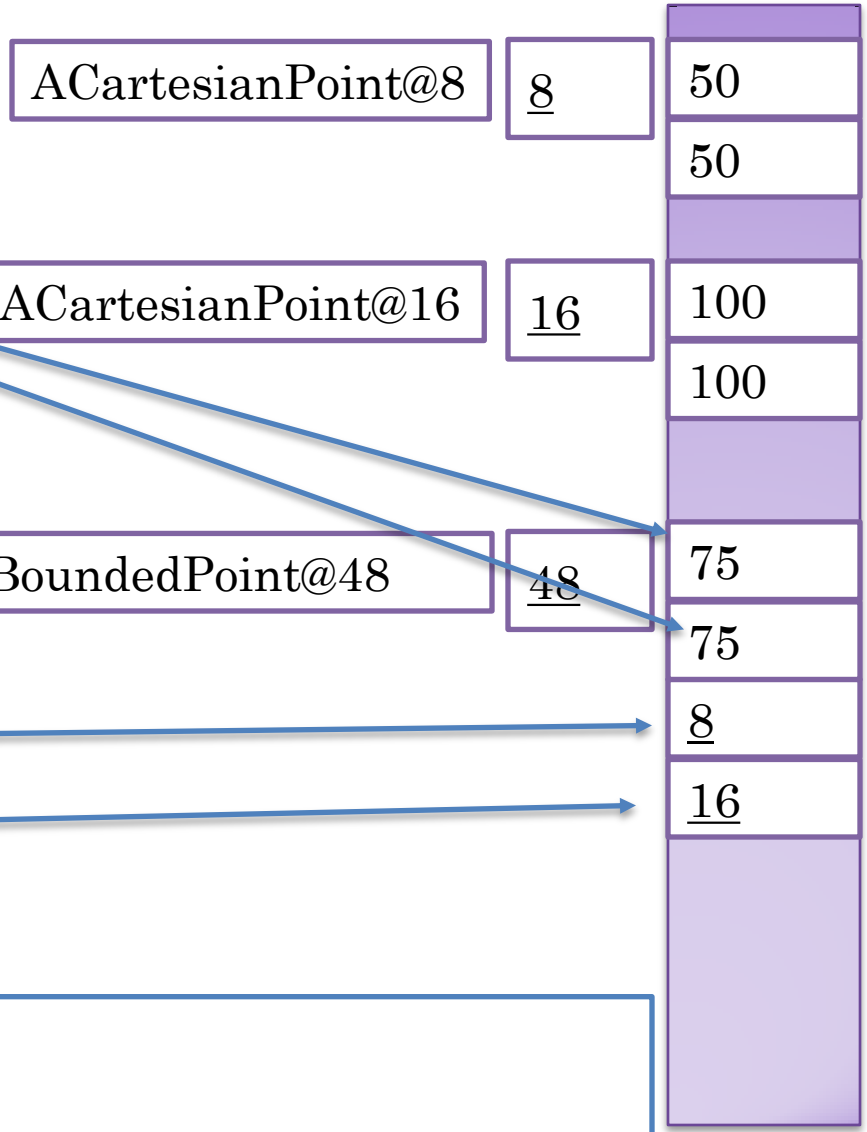
```
public class ABoundedPoint
extends AMutablePoint
implements BoundedPoint {

  MutablePoint  upperLeftCorner ;

  MutablePoint  lowerRightCorner;

    …

}
```




```
new ABoundedPoint(75, 75,
      new AMutablePoint (50,50),
      new AMutablePoint (100,100))
```

Inheritance and Memory Representation

```
public class ACartesianPoint
implements Point {

        int x, y;

    …

}
```

```
public class ABoundedPoint
extends AMutablePoint
implements BoundedPoint {

   int x, y;

   MutablePoint  upperLeftCorner ;

   MutablePoint  lowerRightCorner;

      …

}
```

```
new ABoundedPoint(75, 75,
      new AMutablePoint (50,50),
      new AMutablePoint (100,100))
```