Comp 401
Metamorphosis: Casting vs. Conversion

Instructor: Prasun Dewan
PREREQUISITE

- Interfaces
- Types Math
- Deep vs. Shallow Copy (Optional)
Point point = new ACartesianPoint (50, 100);
point = (APolarPoint) point;

Class cast exception, Java cast does not automatically convert an instance of a class to an instance of another class
It does not even automatically (deep) copy
Ambiguous semantics

Point point = new AMutablePoint (50, 100);
point = (ABoundedPoint) point;

Value of additional properties (upper and lower right corner)?
AS METHODS FOR METAMORPHOSIS

```java
public class AConvertibleCartesianPoint
    extends AMutablePoint implements ConvertiblePoint {
    public AConvertibleCartesianPoint(int theX, int theY) {
        super (theX, theY);
    }
    public ConvertiblePoint asCartesianPoint() {
        return this; // could also clone
    }
    public ConvertiblePoint asPolarPoint() {
        return new AConvertiblePolarPoint(getRadius(), getAngle());
    }
}
```

ConvertiblePoint point =
    new AConvertibleCartesianPoint (50, 100);
point = point.asPolarPoint();

Programmer decides what correct conversion means
**Automatic Conversion of Primitive Values**

Java cast does not automatically convert an instance of a class to an instance of another class.

```java
int intVal = 5;
long longVal = intVal;
double doubleVal = intVal;
longVal = Long.MAX_VALUE;
intVal = (int) longVal;
```

In each of the multi-type assignments, a primitive memory value copied and stored in another primitive memory value of different size.

In the last case, cast forces conversion.
Primitive ➔ Another Primitive

```java
int intVal = 5;
long longVal = intVal;
double doubleVal = intVal;
longVal = Long.MAX_VALUE;
intVal = (int) longVal;
```
PRIMITIVE $\leftrightarrow$ OBJECT

```java
intVal = (int) longVal;
Integer integerVal = intVal;
int intVal2 = integerVal;
```
Wrapper Types

Java automatically converts between primitive and wrapper types when assigning an expression to a variable.
MANUAL WRAPPING AND UNWRAPPING

- **Integer**
  - public Integer(int value)
  - public int intValue()

- **Double**
  - public Double(double value)
  - public double doubleValue()

- **Boolean**
  - public Boolean(boolean value)
  - public boolean booleanValue()

- **Character**
  - public Character(char value)
  - public char charValue()

- **Float, Short, Long**
**Storage of Primitives and Wrapped Values**

```java
int i = 5
double d = 5.5

Integer I = new Integer(i)
Double D = new Double(d)
```

<table>
<thead>
<tr>
<th></th>
<th>int</th>
<th>double</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>5.5</td>
</tr>
</tbody>
</table>

- Different sizes: Integer (8 bytes) vs. Double (16 bytes)
- Same size: int (4 bytes) vs. double (8 bytes)