

Wrapper Classes

Using primitive types with an `ArrayList`

Alwin Tareen

Wrapper Classes

- ▶ Unlike an array, an `ArrayList` can only contain objects, not primitive types.
- ▶ Since numbers are primitive types in Java, you cannot directly insert them into an `ArrayList`.
- ▶ You must turn them into objects by using wrapper classes.

Primitive data type	Wrapper class
<code>int</code>	<code>Integer</code>
<code>double</code>	<code>Double</code>

- ▶ Note that the wrapper class names start with uppercase letters.

Wrapper Classes

- ▶ Each wrapper class object contains a value of the corresponding primitive type.
- ▶ For example, you can create an object of the wrapper class `Double` with the following code:

```
Double num = new Double(29.95);
```

- ▶ The object which is created contains a value of type `double`.
- ▶ Wrapper objects can be used anywhere that objects are required, instead of primitive type values.

Auto-boxing

- ▶ Conversion between primitive types and the corresponding wrapper classes is automatic.
- ▶ This process is called **auto-boxing**.
- ▶ For example, if you assign a number to a `Double` object, the number is automatically, “placed inside a box,” that is, placed inside a wrapper object.

```
Double num = 29.95;
```

- ▶ Wrapper objects are automatically “unboxed” to primitive types.

```
double val = num;
```

ArrayLists of Numbers

- ▶ Placing numbers into an `ArrayList` is straightforward.
- ▶ Simply remember to use the correct wrapper type when you declare the `ArrayList`, and then rely on auto-boxing.

```
ArrayList<Double> scores = new ArrayList<Double>();  
scores.add(29.95);  
double val = scores.get(0);
```

Wrapper Classes: End of Notes