PSOOP-Lec 22: WRAPPER CLASSES IN JAVA

-Compiled by Nikahat Mulla

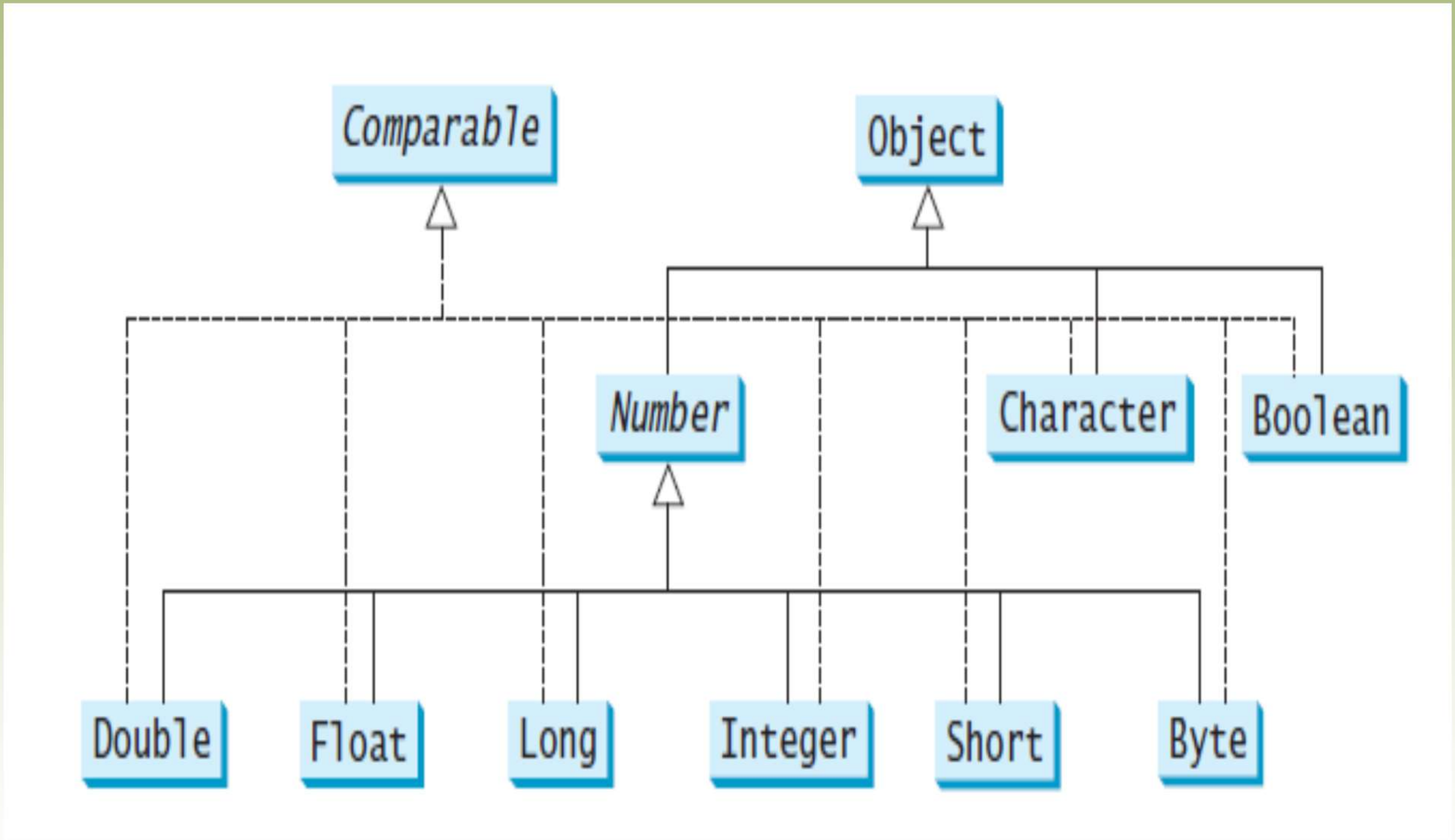
Agenda

- Wrapper Classes

Wrapper Classes

- Java treats objects differently from variables of Primitive types
  - Some times we need to treat int, char, float values as Objects
  - Java provides Wrapper Classes for each primitive type which wraps the value as an Object

Inheritance Hierarchy of Wrapper Classes



Wrapper Classes

- The following declaration creates an `Integer` object which is a reference to an object with the integer value 40

```
Integer age = new
   Integer(40);
```

- An object of a wrapper class is used in situations where a primitive value will not suffice
- For example, some objects serve as containers of other objects
- Primitive values could not be stored in such containers, but wrapper objects could be

Wrapper Classes

- Wrapper classes may contain static methods that help manage the associated type
  - For example, the `Integer` class contains a method to convert digits stored in a `String` to an `int` value:

```
num = Integer.parseInt(str);
```

- Wrapper classes often contain useful constants
  - For example, the `Integer` class contains `MIN_VALUE` and `MAX_VALUE` for the smallest and largest `int` values

Wrapper Classes (Contd…)

- The `java.lang` package contains a *wrapper class* that corresponds to each primitive type:

Primitive Type	**Wrapper Class**	**Constructor Arguments**
byte	Byte	byte or String
short	Short	short or String
int	Integer	int or String
long	Long	long or String
float	Float	float, double or String
double	Double	double or String
char	Character	char
boolean	Boolean	boolean or String

Note: The Wrapper classes do not contain a no-argument constructor

Wrapper Classes

- Converting from primitive to wrapper class is called as Boxing

```
Integer intobj = new Integer(575);
```

- Converting from wrapper class to primitive is called as Unboxing

```
int i = intobj.intValue();
```

Integer

Constructors

Integer(int value)

Integer(String s)

Constants (static)

MAX_VALUE	Maximum positive value
MIN_VALUE	Minimum positive value
TYPE	The Class object for int
SIZE	Number of bits used to represent int type

Integer Class example

Methods in java.lang.Integer

int compareTo(Integer anotherInteger)

double intValue() // similarly doubleValue(), byteValue(), floatValue(), shortValue()

boolean equals(Object obj)

static int parseInt(String s)

static int parseInt(String s, int radix)

static String toBinaryString(int i) // toHexString(), toOctalString()

String toString()

static String toString(int i)

static Integer valueOf(String s)

static Integer valueOf(String s, int radix)

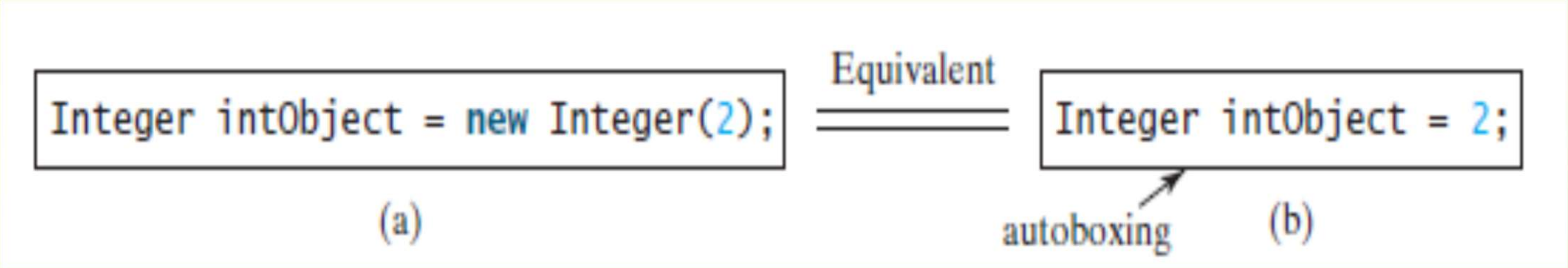
Autoboxing

- *Autoboxing* is the automatic conversion of a primitive value to a corresponding wrapper object:

```
Integer obj;
int num = 42;
obj = num;
```

- The assignment creates the appropriate `Integer` object wrapping a value of 42
- The reverse conversion (called *unboxing*) also occurs automatically as needed

AutoBoxing and unboxing



```
Integer intObject = new Integer(2);
```

```
Integer intObject = 2;
```

```
class AutoBox {
   public static void main(String args[]) {
     Integer iOb = 100;     // autobox an int
     int i = iOb;                      // auto-unbox
     System.out.println(i + " " + iOb);
   }
}
```

displays 100 100

Wrapper Classes (Contd…)

- Wrapper classes have a lot of useful methods

  Examples:

  ```
  Character.toLowerCase(ch)
  Character.isLetter(ch)
  ```

- A common translation is converting a string to a numeric type such as an int

  Example:

  ```
  String s = "65000";
  int i = Integer.parseInt(s);
  ```

Thank You