

Wrapper class Integer

a0

?

234

?

Integer

```

intValue()
byteValue()
shortValue()
longValue()
floatValue()
doubleValue()
toString()
equals(Object)

```

The `int` field can't be changed — it is **immutable**.

```

/** = "ob is a class Integer object, and its
wrapped value equals this object's wrapped
value". */
public static boolean equals(Integer ob)

```

CS101J, Cornell 0

Wrapper class Integer

a0

?

234

?

Integer

```

intValue()
byteValue()
shortValue()
longValue()
floatValue()
doubleValue()
toString()
equals(Object)

```

static components

```

MIN_VALUE
MAX_VALUE

```

The `int` field can't be changed — it is **immutable**.

Reason for wrapper class Integer: to be able to handle an `int` value as an object.

Reason for wrapper class Integer: to have a place to put constants and static methods that deal with `int` values.

CS101J, Cornell 1

Wrapper class Integer

a0

?

234

?

Integer

```

intValue()
byteValue()
shortValue()
longValue()
floatValue()
doubleValue()
toString()
equals(Object)

```

static components

```

MIN_VALUE
MAX_VALUE
toBinaryString(int)
toOctalString(int)
toHexString(int)
parseInt(String)

```

The `int` field can't be changed — it is **immutable**.

Reason for wrapper class Integer: to be able to handle an `int` value as an object.

Reason for wrapper class Integer: to have a place to put constants and static methods that deal with `int` values.

CS101J, Cornell 2

Wrapper class Integer

a0

?

234

?

Integer

```

intValue()
byteValue()
shortValue()
longValue()
floatValue()
doubleValue()
toString()
equals(Object)

```

static components

```

MIN_VALUE
MAX_VALUE
toBinaryString(int)
toOctalString(int)
toHexString(int)
parseInt(String)

```

The `int` field can't be changed — it is **immutable**.

Reason for wrapper class Integer: to be able to handle an `int` value as an object.

Reason for wrapper class Integer: to have a place to put constants and static methods that deal with `int` values.

CS101J, Cornell 3