AUTOBOXING AND UNBOXING

BOOLEAN AND CHARACTER VALUES

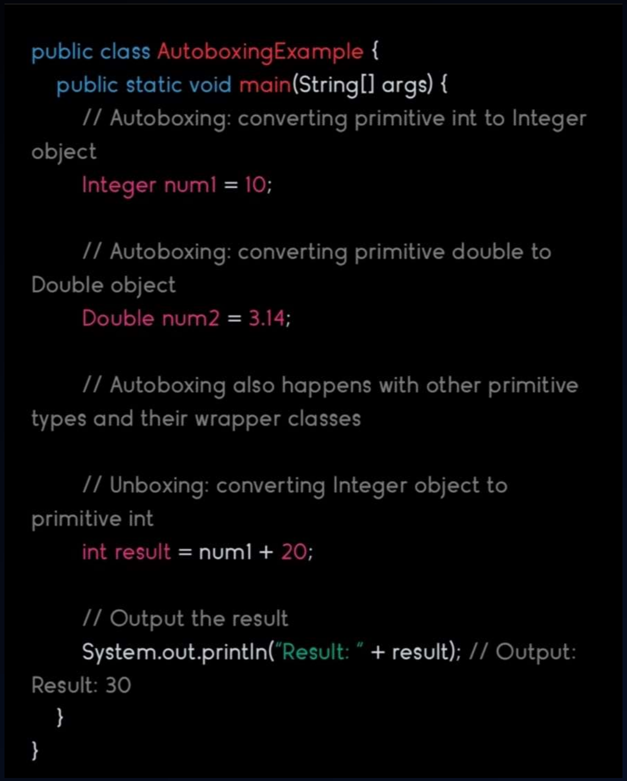
Content

- Autoboxing Boolean Values
- Unboxing Boolean Values
- Autoboxing and Unboxing for Character Values

What is Autoboxing?

- Autoboxing is the automatic conversion that the Java compiler makes between primitive data types and their corresponding wrapper classes.
- For example, converting an int to an Integer or a double to a Double.
- Autoboxing allows you to use primitive data types and their wrapper classes interchangeably in Java code, making it more convenient to work with collections and methods that require objects rather than primitives.

Here's an example of autoboxing in Java:



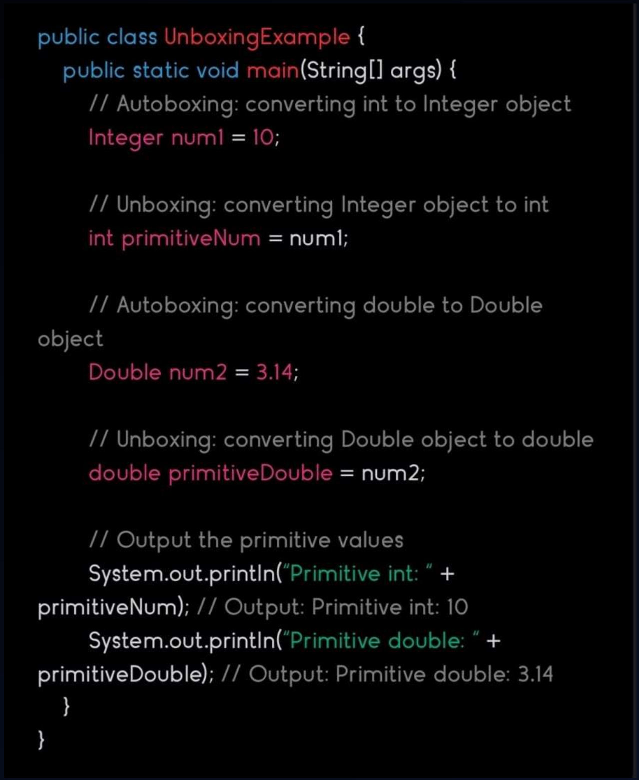
```
public class AutoboxingExample {
   public static void main(String[] args) {
      // Autoboxing: converting primitive int to Integer
object
      Integer num1 = 10;

      // Autoboxing: converting primitive double to
Double object
      Double num2 = 3.14;

      // Autoboxing also happens with other primitive
types and their wrapper classes

      // Unboxing: converting Integer object to
primitive int
      int result = num1 + 20;

      // Output the result
      System.out.println(“Result: ” + result); // Output:
Result: 30
   }
}
```

- In this example, Autoboxing occurs when assigning primitive values 10 and 3.14 to Integer and Double objects respectively.
- Later, unboxing happens when adding the Integer object num1 to the primitive int 20.

What is Unboxing?

- Unboxing is the opposite process of autoboxing in Java. It involves automatically converting a wrapper class object back to its corresponding primitive data type.
- For example, converting an Integer object to an int or a Double object to a double.
- Unboxing allows you to extract the primitive value from a wrapper class object, enabling you to perform operations with primitive data types in Java.

Here's an example demonstrating unboxing in Java:



```
public class UnboxingExample {
   public static void main(String[] args) {
      // Autoboxing: converting int to Integer object
      Integer num1 = 10;

      // Unboxing: converting Integer object to int
      int primitiveNum = num1;

      // Autoboxing: converting double to Double
object
      Double num2 = 3.14;

      // Unboxing: converting Double object to double
      double primitiveDouble = num2;

      // Output the primitive values
      System.out.println("Primitive int: " +
primitiveNum); // Output: Primitive int: 10
      System.out.println("Primitive double: " +
primitiveDouble); // Output: Primitive double: 3.14
   }
}
```

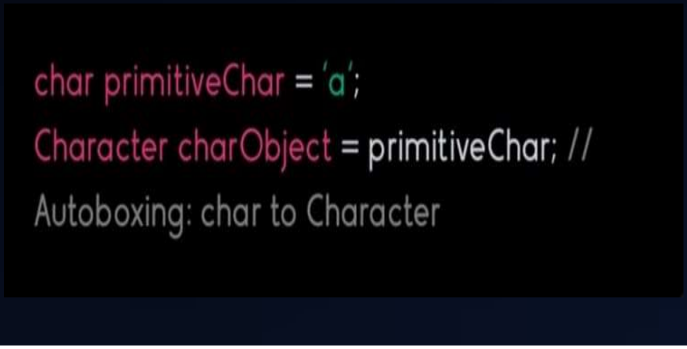
- In this example, unboxing occurs when converting Integer and Double objects (num1 and num2) back to their corresponding primitive types (int and double).
- This allows us to use the primitive values in calculations or other operations

Autoboxing and Unboxing for Character Values

Autoboxing:

When you assign a primitive ‘char’ value to a variable of type ‘Character’, autoboxing automatically converts the ‘char’ value to a ‘Character’ object.

For example:

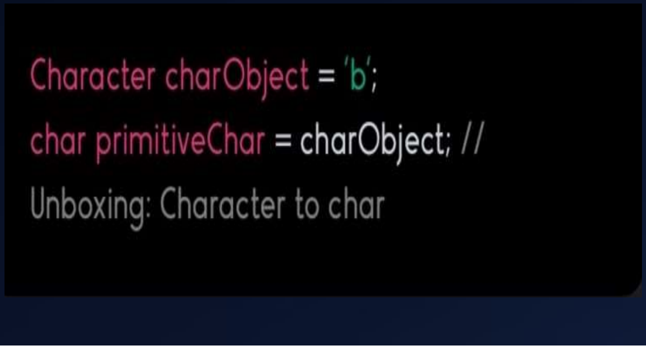


```
char primitiveChar = 'a';
Character charObject = primitiveChar; //
Autoboxing: char to Character
```

Unboxing:

When you assign a ‘Character’ object to a variable of type ‘char’, unboxing automatically extracts the primitive ‘char’ value from the ‘Character’ object.

For example:



```
Character charObject = 'b';
char primitiveChar = charObject; //
Unboxing: Character to char
```