

CS2110: What's a wrapper class

Introduction

We cannot give you a lecture on everything in Java; some simple concepts and material you have to pick up on your own. While we will probably mention wrapper classes in a recitation, you may need the concept before the recitation. But the idea is simple and you can pick it up easily. Here are some resources:

1. On the resources page of the 2110 website are links to ppt slides (and a pdf file of them) that summarizes Java. Always look there for help! The table of contents tells you to look on slide 46.
2. Look in the course text for “wrapper class”
3. Type wrapper class Java into a search engine and you will find many tutorials.
4. Look in the Java API specs for class Integer.

Wrapper classes in a nutshell

In creating a new LinkedList of **ints**, we cannot use `new LinkedList<int>()` because the name between ‘<’ and ‘>’ has to be a class name, not a primitive type. Java’s solution to this problem is to have a class Integer—in package `java.lang`, so you don’t have to import it— each object of which wraps (or contains) an `int`.

The word *wraps* and the phrase *wrapper class* are Java’s, not ours. They make sense. Below are a bunch of wrappers, the last of which is an object of class Integer. In order, these wrappers wrap a sandwich, a cupcake, spring rolls, and an **int**.



Like Strings, Integers are immutable. You can’t change the **int** that is wrapped in an Integer object

Java makes it easy to go back and forth between **int** and Integer:

```
Integer d= 5; // Java automatically wraps the 5 in a new Integer object and stores a pointer to it in d
// Java calls this autoboxing. Autowrapping would have been better.

ll.prepend(4); // Assuming ll is a LinkedList, Java will automatically autobox the 4 as above.

int k= d; // Java automatically unboxes (unwraps is better) the int in Object d
```

Java will autobox and autounbox for you in most situations where you would want it to.

Class Integer has a constructor `Integer(int)` and a bunch of instance methods, like `d.equals(d1)`, `d.intValue()`, `d.toString()` .

Class Integer has some static variables, like `Integer.MAX_VALUE` and `Integer.MIN_VALUE` .

Class Integer has some static methods, like `Integer.parseInt(String s)` .

Each primitive type has its wrapper class, defined in package `java.lang` so you don’t have to import it explicitly. Each has instance methods, static variables, and static methods appropriate to that type. Here they are:

primitive type	wrapper class
byte	Byte
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean