



## Constraining the elements of an ArrayList


 You now know that you can create an instance of class `java.util.ArrayList` and save it using an assignment like

```
ArrayList b= new ArrayList();
```

The elements of this list have the apparent class `Object`. When an element is added to the list, it is automatically promoted to `Object`.

 Beginning in Java 1.5, you can constrain the objects of a list by placing a class name, in angular brackets (the less-than and greater-than symbols) after the class name `ArrayList`.

```
b= new ArrayList<Integer>();
```

 Now, only objects of class `Integer` can be added to `b`. An attempt to add an object of a class that is not `Integer` or a subclass of `Integer` will be flagged as illegal, and the program won't compile.

You can put any class in place of `Integer`.

The ability to constrain `ArrayLists` in this fashion simplifies some coding and provides more type safety. By using a constrained `ArrayList`, you can reduce the chance of making an error by adding something to an `ArrayList` that does not belong there. Constrain your `ArrayLists` where possible.