Generic classes can be bounded, limiting the types that can use it.

On this slide, I'm showing the code from my class.

This extends keyword doesn't have the same meaning as extends, when it's used in a class declaration.

This isn't saying our type T extends Player, although it could.

This is saying the prameterized type T, has to be a Player, or a **subtype** of Player.

Now Player in this case could have been either a class or an interface, the syntax would be the same.

This declaration establishes what is called an **upper bound**, on the types that are allowed to be used with this class

```
public class Team<T extends Player> {
```

Why specify an upper bound?

An upper bound permits access to the bounded type's functionality.

An upper bound limits the kind of type parameters you can use when using a generic class. The type used must be equal to, or a subtype of the bounded type.