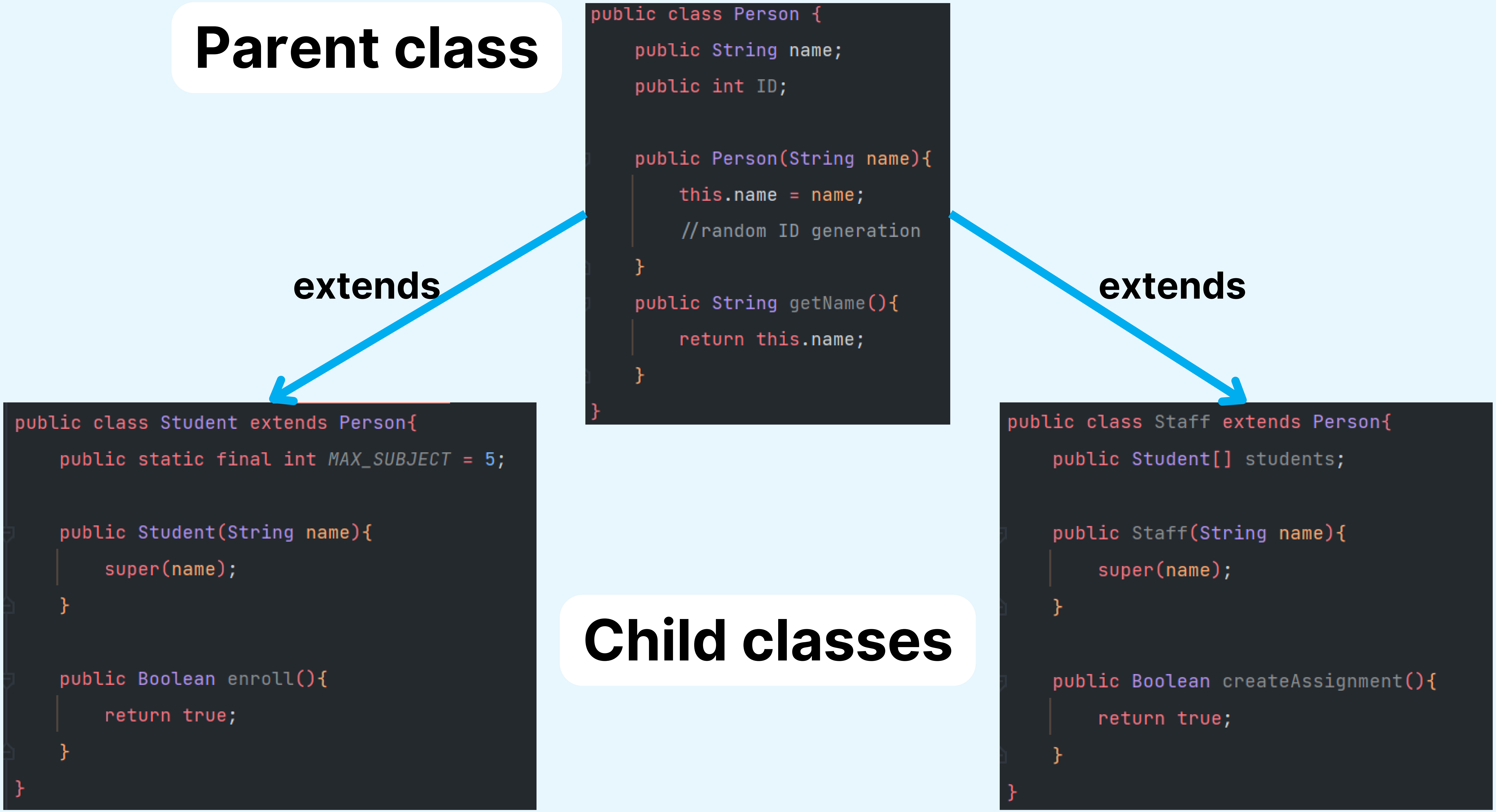
Object Oriented Software Development

Week 6

Inheritance

Child classes inherit from a parent class

Have their attributes, methods, plus their own



```java
public class Person {
    public String name;
    public int ID;

    public Person(String name){
        this.name = name;
        //random ID generation
    }
    public String getName(){
        return this.name;
    }
}
```

```java
public class Student extends Person{
    public static final int MAX_SUBJECT = 5;

    public Student(String name){
        super(name);
    }

    public Boolean enroll(){
        return true;
    }
}
```

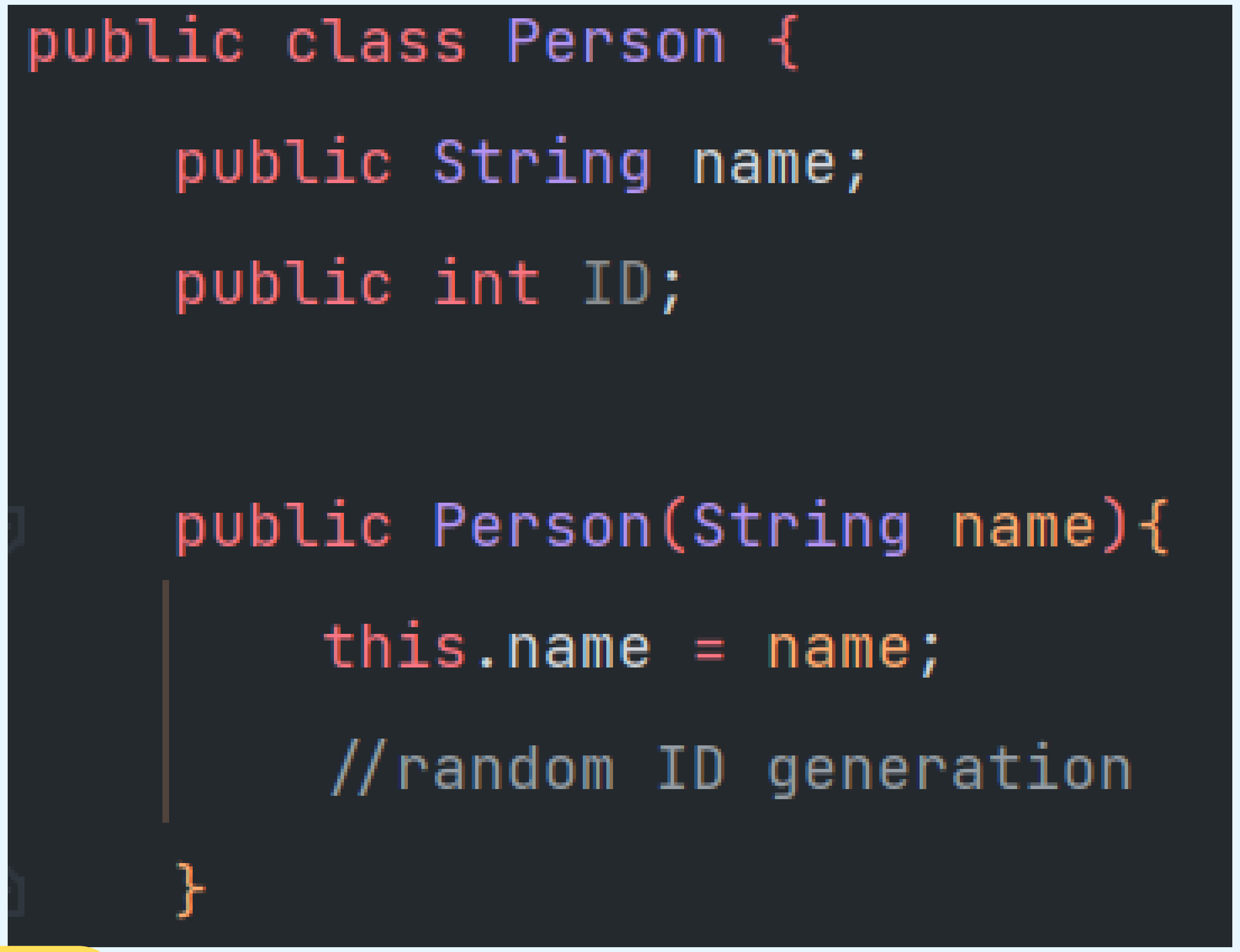
```java
public class Staff extends Person{
    public Student[] students;

    public Staff(String name){
        super(name);
    }

    public Boolean createAssignment(){
        return true;
    }
}
```

Inheritance

Child classes must use super() in their constructor

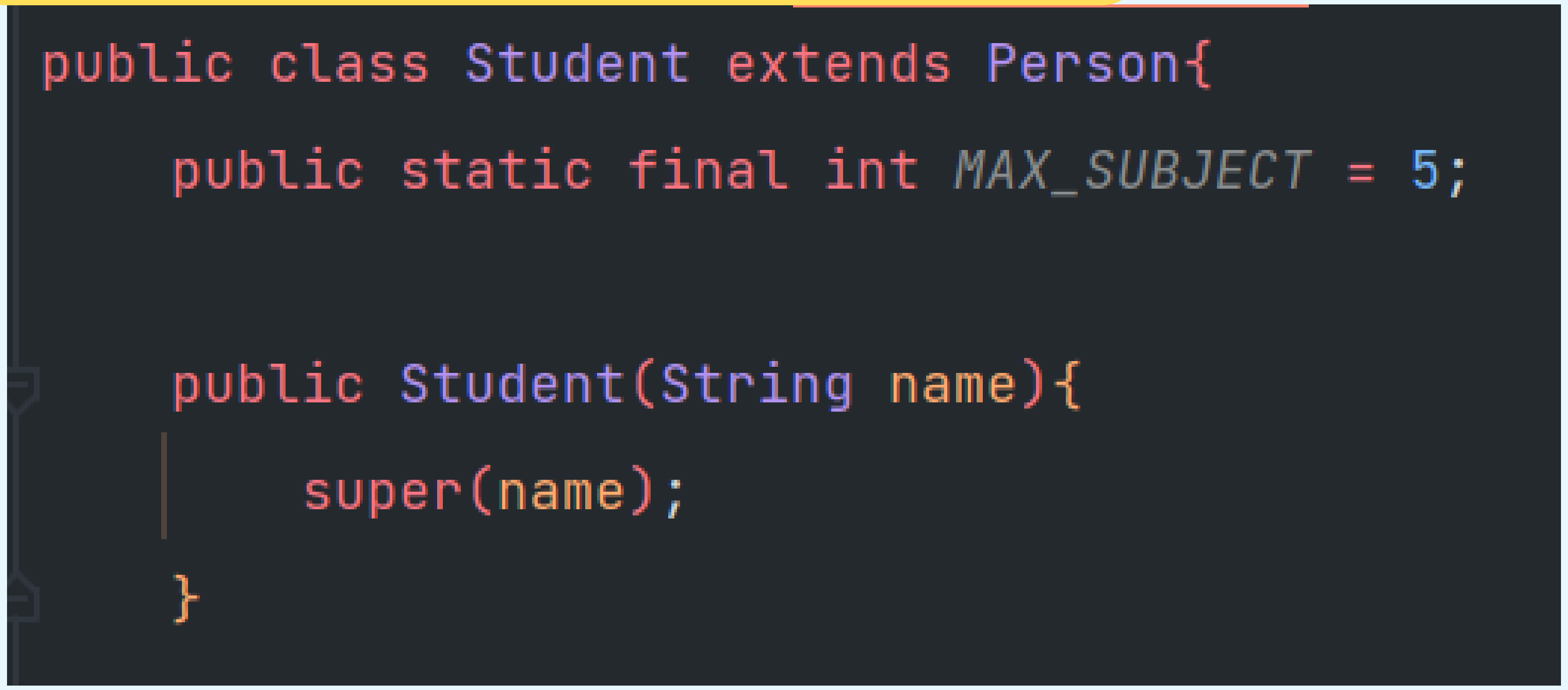
super() calls the parent class constructor

Parent class



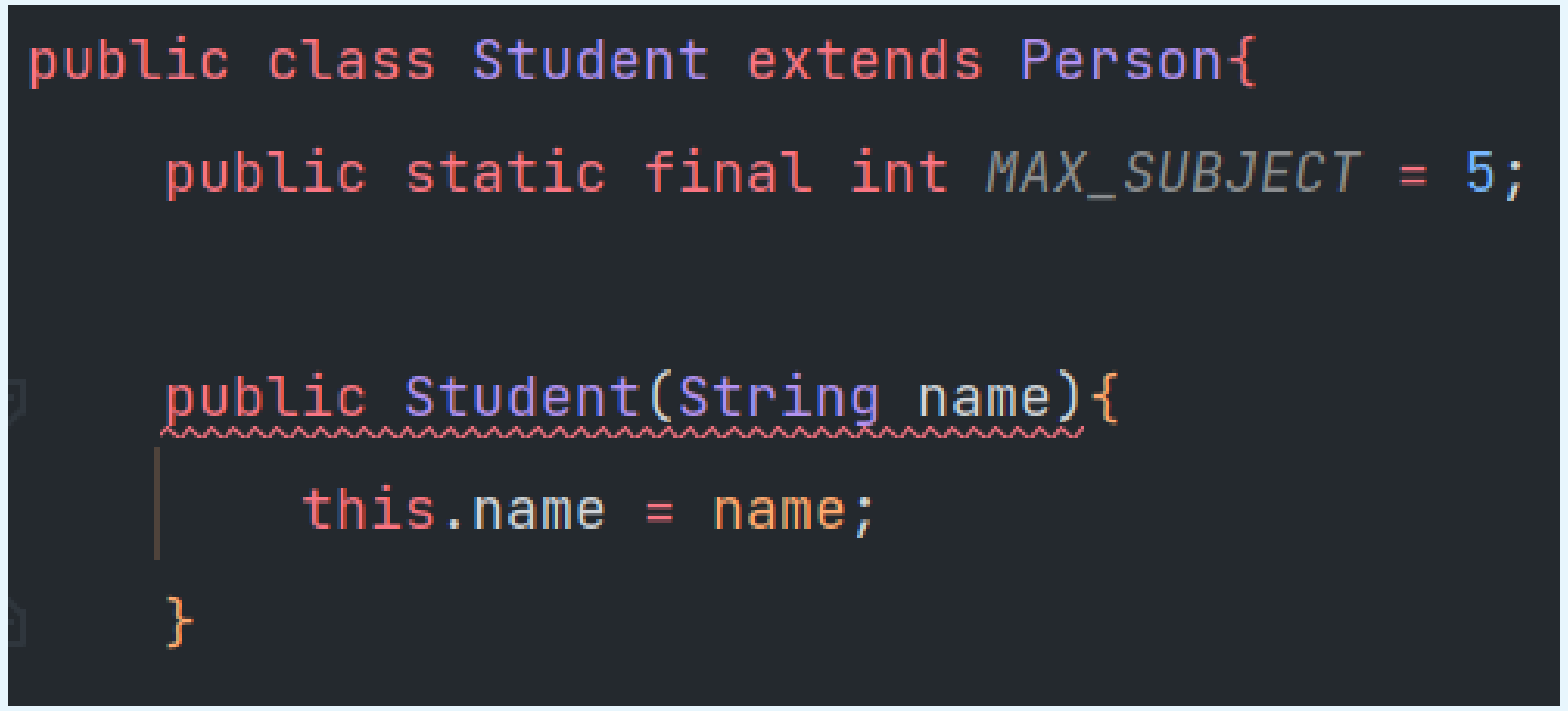
```java
public class Person {
    public String name;
    public int ID;

    public Person(String name){
        this.name = name;
        //random ID generation
    }
```

super() in constructor



```java
public class Student extends Person{
    public static final int MAX_SUBJECT = 5;

    public Student(String name){
        super(name);
    }
```

no super() in constructor



```java
public class Student extends Person{
    public static final int MAX_SUBJECT = 5;

    public Student(String name){
        this.name = name;
    }
```

Upcasting and Downcasting

Changing between parent and child class

```
Person person = new Student("Catie");
```

Upcasting (child -> parent)

```
Student student= (Student) person;
```

Downcasting (parent -> child)

Inheritance

Child classes can override methods in the parent class

use super() to distinguish

Abstract classes

Abstract classes can have **abstract or concrete** methods

Concrete classes can **only have concrete** methods

Does not make sense for a class that can be instantiated (i.e. a concrete class) to have non-instantiable methods

Kahoot!

Pairs or individually

Share a single device

Initial Submission

Due Wednesday April 17th

TAKES 5 MINS DO IT NOW

There will (probably) be NO extensions on this

Initial Submission

1. **Clone** the project repo to your device
2. Download the **skeleton code**
3. **Unzip** that folder
4. Copy the folder **CONTENTS** (not the entire folder)
5. Paste into your **local project repo**
6. Open **terminal**, **cd** into location of the project repo
7. **git add .**
8. **git commit** -m "intial commit"
9. **git push**
10. Profit