Video Name: UMLCode

Topics:
- translating UML diagram to code

Java Class(es): uml package, many classes

```
public class Customer {
    protected String name;
    protected String email;
    // Notice that composition implies a data variable
    private ArrayList<Invoice> invoices;
}
```

```
public class Student extends Customer
```

```
public void purchase(DiscountType discount)
```
Interface and Implementation

```java
public interface Drawable {
    public void draw();
}

public class Flooring implements Drawable {
    @Override
    public void draw() {
    }
}
```

Composition and Aggregation (no difference in code, both imply an instance variable)

```java
public class Room {
    // multiplicity of n
    private ArrayList<Furniture> furnishings;
    // multiplicity of 1
    private Flooring floor;
}
```

NOTE: association would also just be an instance variable. Aggregation and Composition mean that one class “owns” the other. Association is a relationship between two independent objects. These all just translate to instance variables in code. The distinctions are made so that humans looking at the UML might better understand the system design.