5.6 - Class Relationships

- Three of the most common relationships:
  - Dependency: A uses B
  - Aggregation: A has-a B
  - Inheritance: A is-a B

- A dependency exists when one class relies on another, usually by invoking the methods of the other.
- An aggregate is an object that is made up of other objects.
- Therefore aggregation is a has-a relationship
  - A Student has an Address

- Inheritance, later.

5.5 - Static Class Methods

```java
class Helper {
    public static int cube (int num){
        return num * num * num;
    }
}
```

Because it is declared as static, the method can be invoked as:

```java
value = Helper.cube(5);
```

- static associates the method (or var.) with the class rather than with an instance of that class (that’s why they are also called class methods or class vars)
- For example, the methods of the Math class are static:

```java
result = Math.sqrt(25)
```

- Determining if a method or variable should be static is an important design decision.

5.8 - Method Overloading

- Method overloading is the process of giving a single method name multiple definitions
- The compiler determines which method is being invoked by analyzing the parameters
- The signature of each overloaded method must be unique
- The signature includes the number, type, and order of the parameters, but not the return type!

```java
float tryMe(int x)
{
    return x + .375;
}
float tryMe(int x, float y)
{
    return x * y;
}
```

Invocation

```java
result = tryMe(25, 4.32)
```