

Pro Dev Session



Pro Dev Session

You may want to write this down.

Writing Professional Emails

- Include all appropriate parties
- Make the title clear and searchable
- Include relevant background information
- Include a call to action
- Thank everyone for their involvement
- Check spelling and do not use text abbreviations
- Consider a TL;DR

Pro Dev Session

You may want to write this down.

To: it@company.com
Cc: hiringManager@company.com, newHire@company.com
Subject: Corrected Computer Specs for J NewHire

Dear IT Team,

I'm afraid I sent the incorrect specs for Jan NewHire's laptop. She requires 32GB of RAM rather than the 16GB that I had originally requested.

Please advise Jan (cc'd on this message) on the process for returning her current laptop and on the ETA for the replacement.

I apologize for any inconvenience that my mistake may have caused and appreciate your help in remedying this situation.

For clarity the correct specs are: 512GB SSD, i7+ processor, 32GB RAM.

Thanks you again for your help,
J. Brooks

Pro Dev Session

You may want to write this down.

To: it@company.com

Cc: hiringManager@company.com, newHire@company.com

← CC interested parties

Subject: Corrected Computer Specs for J NewHire

Dear IT Team,

I'm afraid I sent the incorrect specs for Jan NewHire's laptop. She requires 32GB of RAM rather than the 16GB that I had originally requested.

Please advise Jan (cc'd on this message) on the process for returning her current laptop and on the ETA for the replacement.

I apologize for any inconvenience that my mistake may have caused and appreciate your help in remedying this situation.

For clarity the correct specs are: 512GB SSD, i7+ processor, 32GB RAM.

Thanks you again for your help,
J. Brooks

Pro Dev Session

You may want to write this down.

To: it@company.com

Cc: hiringManager@company.com, newHire@company.com

Subject: Corrected Computer Specs for J NewHire

← Clear searchable title

Dear IT Team,

I'm afraid I sent the incorrect specs for Jan NewHire's laptop. She requires 32GB of RAM rather than the 16GB that I had originally requested.

Please advise Jan (cc'd on this message) on the process for returning her current laptop and on the ETA for the replacement.

I apologize for any inconvenience that my mistake may have caused and appreciate your help in remedying this situation.

For clarity the correct specs are: 512GB SSD, i7+ processor, 32GB RAM.

Thanks you again for your help,
J. Brooks

Pro Dev Session

You may want to write this down.

To: it@company.com
Cc: hiringManager@company.com, newHire@company.com
Subject: Corrected Computer Specs for J NewHire

Dear IT Team, ← Greeting

I'm afraid I sent the incorrect specs for Jan NewHire's laptop. She requires 32GB of RAM rather than the 16GB that I had originally requested.

Please advise Jan (cc'd on this message) on the process for returning her current laptop and on the ETA for the replacement.

I apologize for any inconvenience that my mistake may have caused and appreciate your help in remedying this situation.

For clarity the correct specs are: 512GB SSD, i7+ processor, 32GB RAM.

Thanks you again for your help,
J. Brooks

Pro Dev Session

You may want to write this down.

To: it@company.com
Cc: hiringManager@company.com, newHire@company.com
Subject: Corrected Computer Specs for J NewHire

Relevant information

Dear IT Team,

I'm afraid I sent the incorrect specs for Jan NewHire's laptop. She requires 32GB of RAM rather than the 16GB that I had originally requested.

Please advise Jan (cc'd on this message) on the process for returning her current laptop and on the ETA for the replacement.

I apologize for any inconvenience that my mistake may have caused and appreciate your help in remedying this situation.

For clarity the correct specs are: 512GB SSD, i7+ processor, 32GB RAM.

Thanks you again for your help,
J. Brooks

Pro Dev Session

You may want to write this down.

To: it@company.com
Cc: hiringManager@company.com, newHire@company.com
Subject: Corrected Computer Specs for J NewHire

Dear IT Team,

I'm afraid I sent the incorrect specs for Jan NewHire's laptop. She requires 32GB of RAM rather than the 16GB that I had originally requested.

Call to Action

Please advise Jan (cc'd on this message) on the process for returning her current laptop and on the ETA for the replacement.

I apologize for any inconvenience that my mistake may have caused and appreciate your help in remedying this situation.

For clarity the correct specs are: 512GB SSD, i7+ processor, 32GB RAM.

Thanks you again for your help,
J. Brooks

Pro Dev Session

You may want to write this down.

To: it@company.com
Cc: hiringManager@company.com, newHire@company.com
Subject: Corrected Computer Specs for J NewHire

Dear IT Team,

I'm afraid I sent the incorrect specs for Jan NewHire's laptop. She requires 32GB of RAM rather than the 16GB that I had originally requested.

Please advise Jan (cc'd on this message) on the process for returning her current laptop and on the ETA for the replacement.

Thanks

I apologize for any inconvenience that my mistake may have caused and appreciate your help in remedying this situation.

For clarity the correct specs are: 512GB SSD, i7+ processor, 32GB RAM.

Thanks you again for your help,
J. Brooks

Pro Dev Session

You may want to write this down.

Why are we doing this?

- Being a great developer necessarily involves being a great communicator. In this course, we will focus on improving several forms of communication. We've already talked a little about spoken communication in the form of giving feedback, today we'll discuss written communication in the form of professional emails.

What if I already know this stuff?

- Regardless of your level of expertise, humility is an attribute that everyone appreciates. Look for areas that you can improve in your written communication.

Pair Practice

INDEPENDENT PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Write an internal email (including all cc's and subject) expressing that a project you are working on is at risk because the client has not responded with the updated requirements
- Write an external email (including all cc's and subject) expressing that a project you are working on is at risk because the client has not responded with the updated requirements

Your team structure is as follows: 5 developers led by a lead architect report to you. You report to the director of engineering. Director of engineering reports to the President who reports to the CEO. Sales and customer relations also answer to the President. Sales owns the client relationship through the contract signing, after which customer relations owns the account. Kim sold this account and Jamie is the customer relations rep assigned to this account.



**10
minutes!**

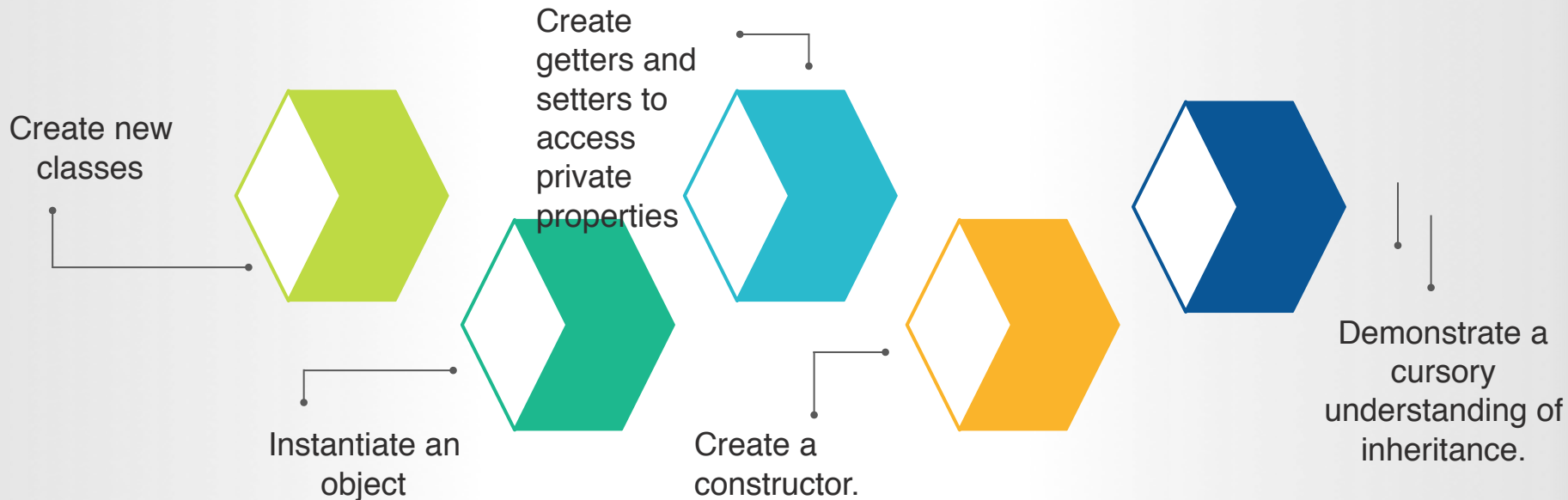
Stand Up



Objectives & Key Outcomes

THE TAKEAWAYS FROM THIS CLASS

By the end of class today, you will be able to:



Objectives & Key Outcomes

THE TAKEAWAYS FROM THIS CLASS

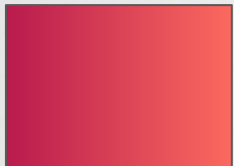


Let's Do This!



A Quick Reminder

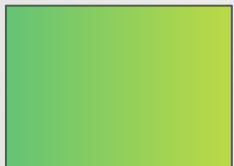
COLOR SYSTEM



Red means stop. When you see a red header, close your laptop.



Yellow means wait. Open your laptop and code along with me.



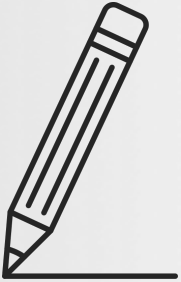
Green means go. This is independent practice time so get to coding!



Classes and Objects

Classes & Objects

BUILDING BLOCKS OF APPLICATIONS



Notebooks Ready? It's time for a mini lecture.



Classes & Objects

BUILDING BLOCKS OF APPLICATIONS

You've learned the building blocks of programming (variables, loops, conditionals, operators, ect) . Now it's time to learn the building blocks of applications: Classes and Objects

Classes and Objects are the cornerstones of Object Oriented Programming.

This might feel a little theoretical for a while but we'll learn all about the practical applications next week.



Classes & Objects

BUILDING BLOCKS OF APPLICATIONS


What is Object Oriented Programming?

- It's a style of application architecture.

Are all applications built using OOP?

- Nope. But a lot are.

What does all this mean?

- Building an application is a lot like building a house. There are several styles of framing for houses: Steel framing, Wood framing, and Concrete Framing. Each style requires the workers to know the material, the tools, and the overarching principles.
 - Some workers may practice both wood and steel framing, but they need some level of mastery in one before learning the other.
 - OOP is like the wood framing of development. It's trusted. It's widely used. It's worth mastering.
- 

Classes & Objects

BUILDING BLOCKS OF APPLICATIONS

So what is a class exactly?

A class is like a blueprint from which you can create objects.



Classes & Objects

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
public class Student{  
    public String name;  
    public int id;  
    public double gpa;  
}
```

We've created a blueprint for a student. All students have a name, gpa, and id.



Classes & Objects


WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
public static void main(String[] args){  
    Student student1 = new Student();  
    String favMonkey = "spider monkey";  
}
```

The `new` keyword is used to create a new student object from the `Student` class. This student has its own copy of `name`, `id`, and `gpa`.

`Student` is just a custom type of variable. Note the similarities between a `String` and a `Student`. The only real difference is that someone else made a hidden `String` class for us.



Classes & Objects


WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
public static void main(String[] args){  
    Student student1 = new Student();  
    student1.name = "Sheryl";  
    System.out.println(student1.name);  
}
```

dot notation allows us to access specific properties on an object... but there's a serious issue here. What if some other part of the application wants to come in and set the gpa to 200?

That should NOT be allowed. We need to find a way to control access to Student properties.



Classes & Objects

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
public class Student{  
    private String name;  
    private int id;  
    private double gpa;  
}
```

Now no other part of the application can access these properties. We need some way to give some level of access without giving full access.

Classes & Objects

CODE-A-LONG

Open your laptop. Code with me. Don't jump ahead.

```
public class Student{  
    private String name;  
    private int id;  
    private double gpa;  
  
    public String getName(){  
        return this.name;  
    }  
  
    public void setName(String name){  
        this.name = name;  
    }  
}
```

Classes & Objects

PAIR PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Add getters and setters for id and gpa.
- Ensure that you can set and get these properties from the main method
- CHALLENGE: Add logic to the gpa setter so that the maximum gpa is 4.0. This means that if someone passes in 15, the gpa should be set to 4.0. If someone passes in 3.2, then the gpa should be 3.2.



**15
minutes!**

Classes & Objects

BUILDING BLOCKS OF APPLICATIONS

Getters and setters aren't the only methods we need.

There are surely more actions that all Students share. Perhaps all students should be able to register for classes.



Classes & Objects

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
public class Student{  
    String name;  
    int id;  
    double gpa;  
    int classCount;  
    boolean isEnrolled;  
  
    public void register(){  
        this.classCount++;  
        this.isEnrolled = true;  
    }  
}
```

Classes & Objects

PAIR PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Add an unregister method to Student class. This method should reduce the classCount by 1. If the resulting classCount is less than 1, toggle isEnrolled to false.

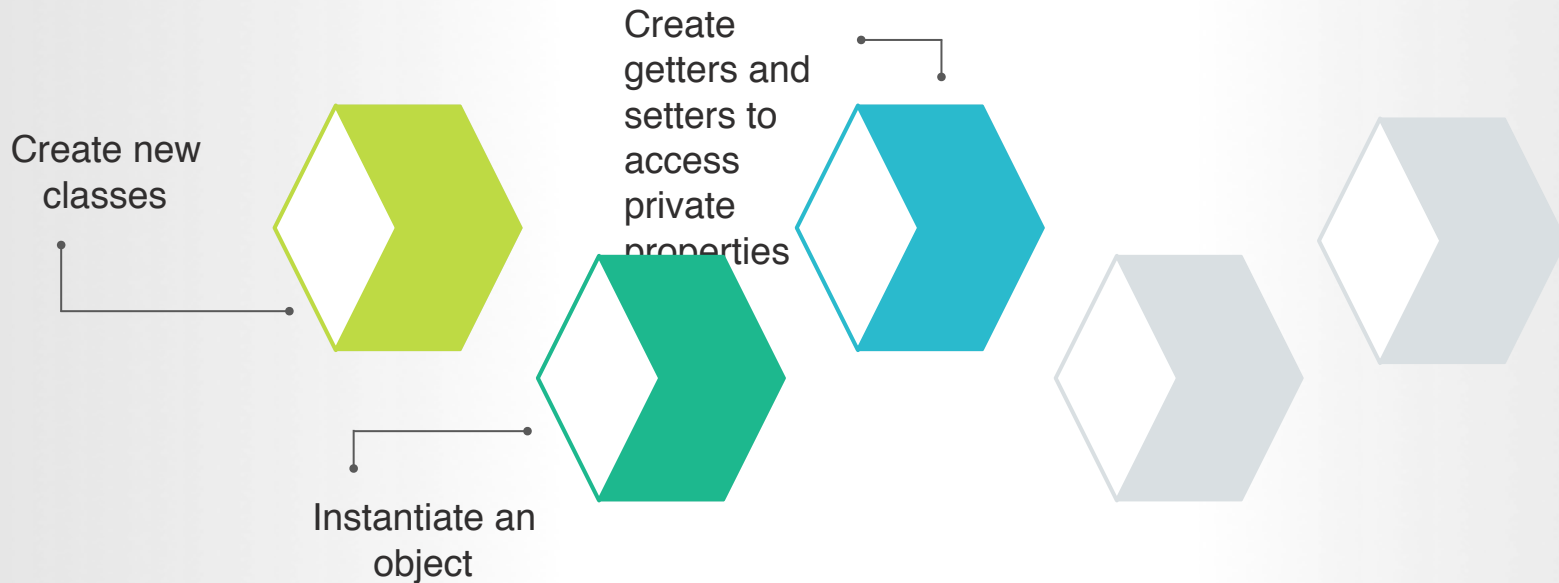


**10
minutes!**

Objectives & Key Outcomes

THE TAKEAWAYS FROM THIS CLASS

By the end of class today, you will be able to:





Stay Seated & Take 3 Deep Breaths.

RELAX.

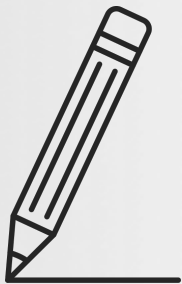
Now take a short walk. Clear your head. After a few minutes break, quickly review your notes.
We'll start back in 5 minutes.



Object Oriented Programming

OOP

OBJECT ORIENTED PROGRAMMING



Notebooks Ready? It's time for a mini lecture.



OOP

OBJECT ORIENTED PROGRAMMING

Object Oriented Programming means using classes and objects created from those classes to create an application. So you've made a Student objects and Animal Objects, but why would you ever do this and how do you build an application using Students and Animals?

To understand that, let's zoom back out and recall how a web application works.



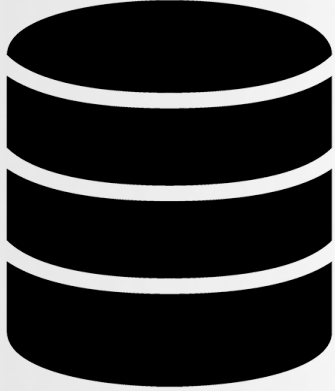
NOTE: We are about to take a short detour. It's okay if you don't get all of this. This is just the first of many times that we'll explain the role of OOP in creating web applications.



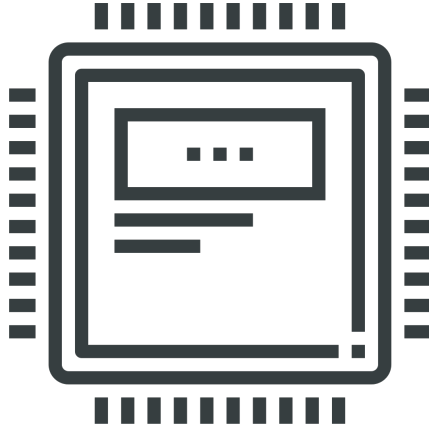
How the Web Works

ANATOMY OF A WEB APPLICATION

Web applications are composed of 3 major components



Database



Server

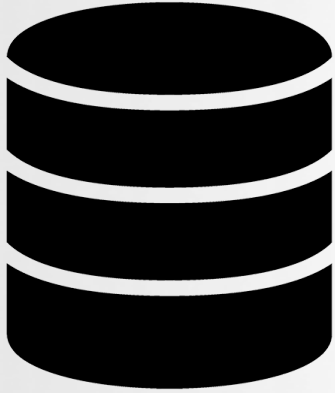


Client

How the Web Works

DATABASES

Databases store data. On a social media site, this might be your username, password, friends, birthday, posts, etc.



Database



Server

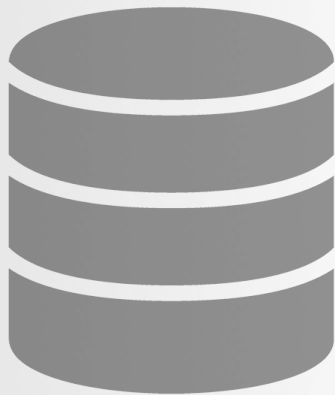


Client

How the Web Works

CLIENTS

Clients are the computers rendering the web app. Like your computer. My computer. The computer of that person sitting next to you.



Database



Server



Client

How the Web Works

SERVERS

Servers manage the flow of communication between the client and the server. Servers are how the data from the database gets to page you are viewing.



Database



Server



Client

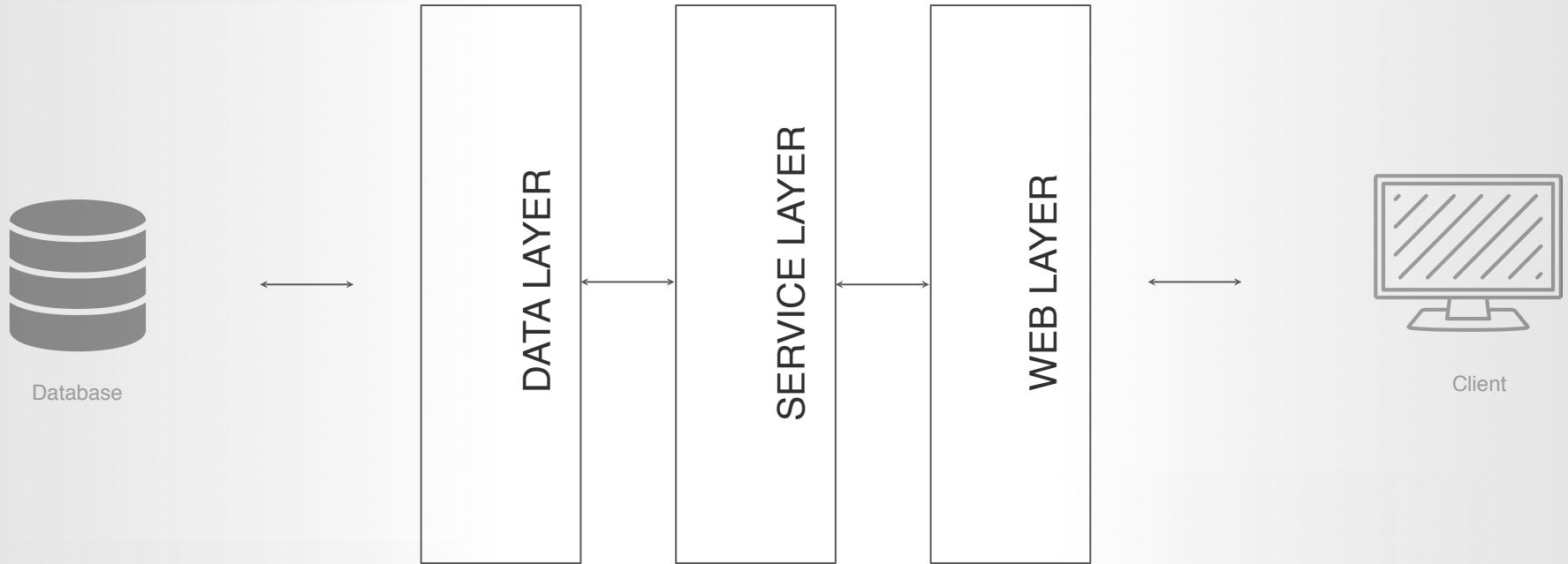
Let's look ahead a little, to understand this more deeply...



Server Structure

ANATOMY OF A SERVER

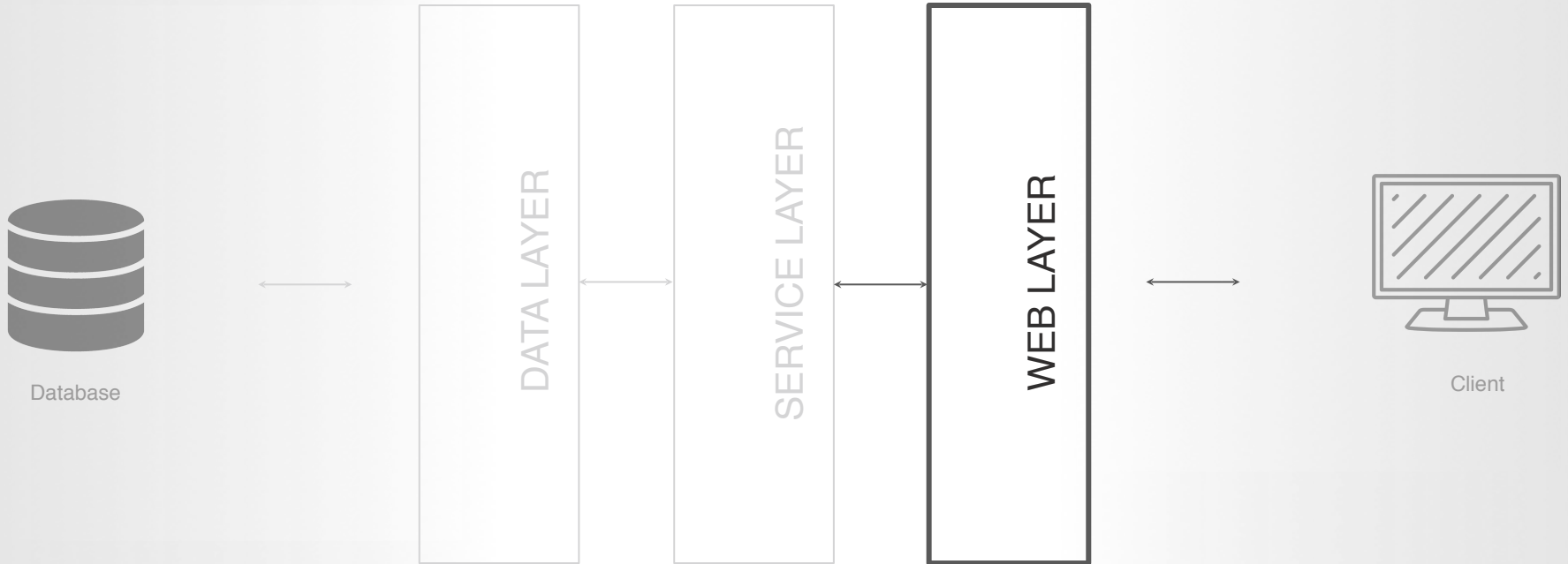
As Java developers, we're focussed on building servers. Spring servers have 3 parts:



Server Structure

WEB LAYER

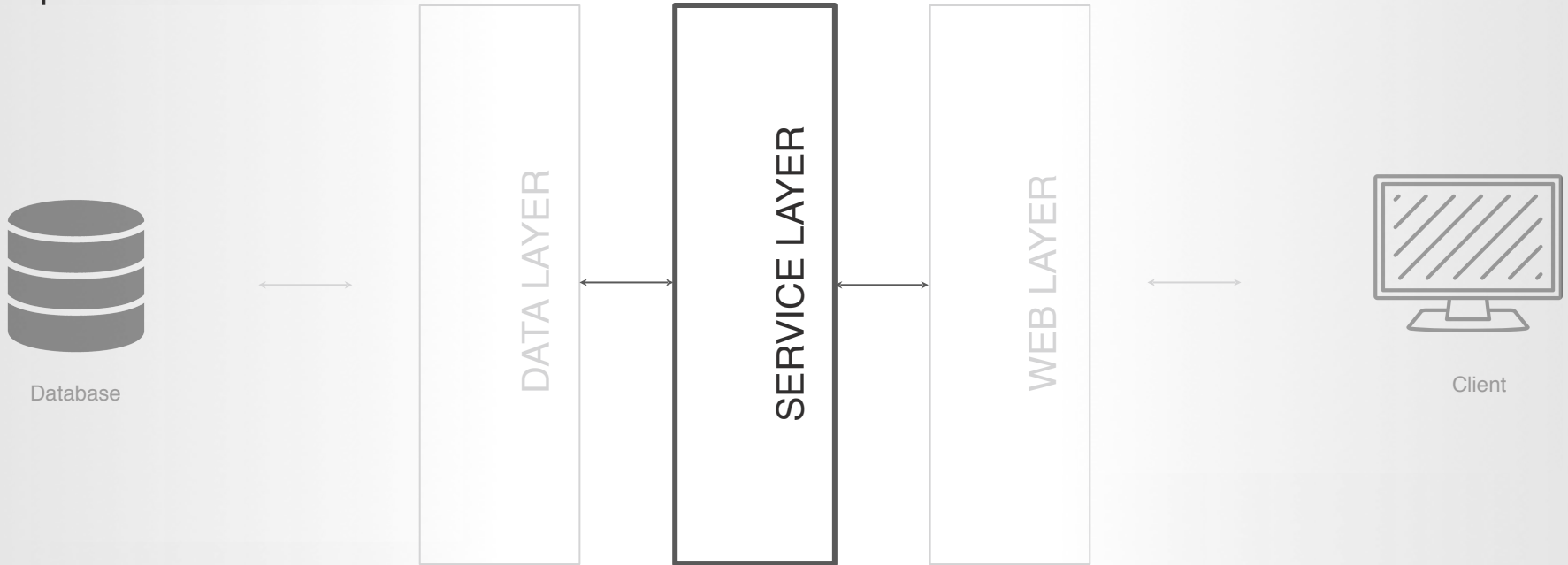
The web layer is responsible for processing requests from the client and sending responses.



Server Structure

SERVICE LAYER

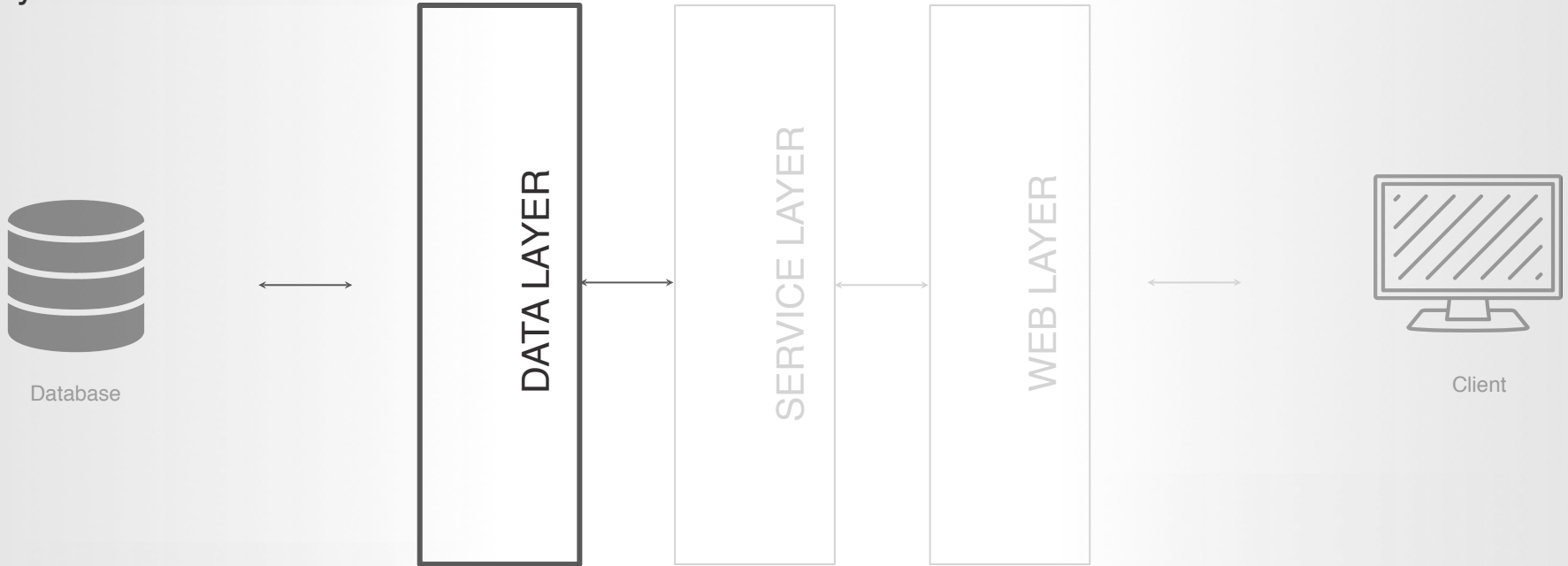
The service layer is responsible for any logic that needs to be performed on the request or the response.



Server Structure

DATA LAYER

The data layer is responsible for getting data from the database and sending it to the service layer.

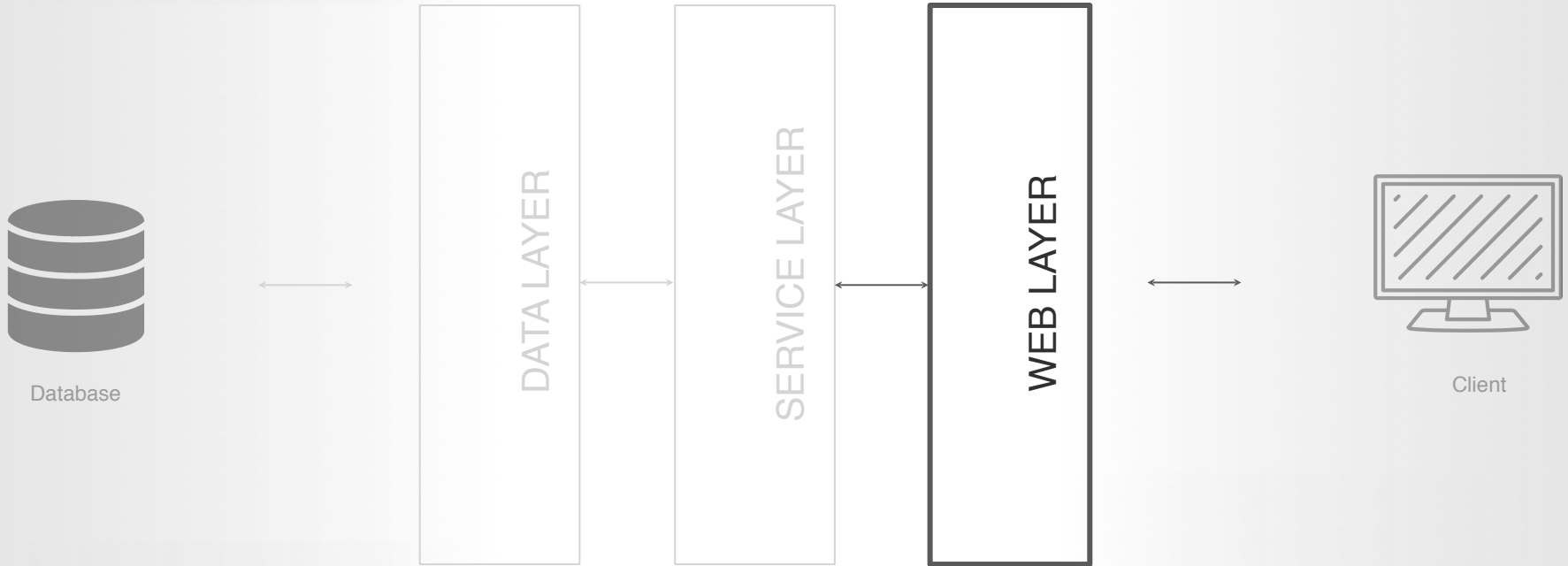


Let's look at a more concrete example. Suppose we are running a movie streaming service and a user (let's call her Lucy) clicks on the recommendations page



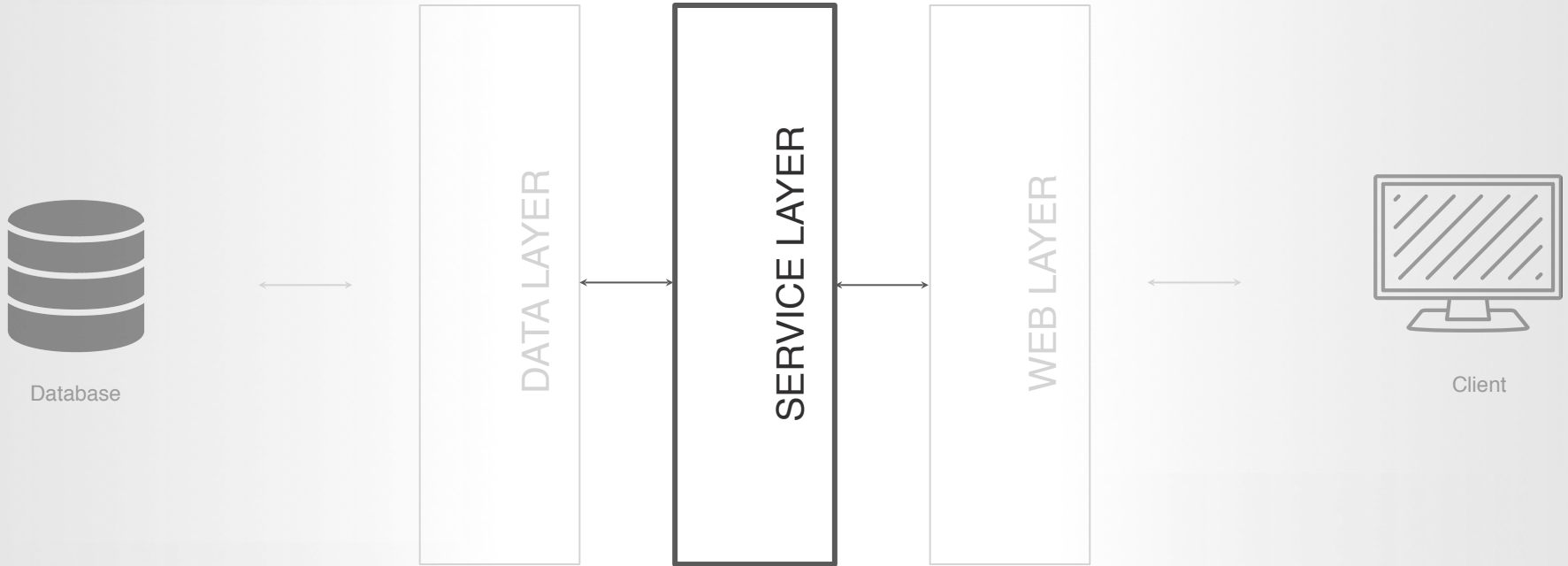
Server Structure

The web layer is responsible for receiving the request for recommended movies and calling the proper method in the service layer, then it waits to get a response from the service layer and sends this response back to Lucy's computer (client).



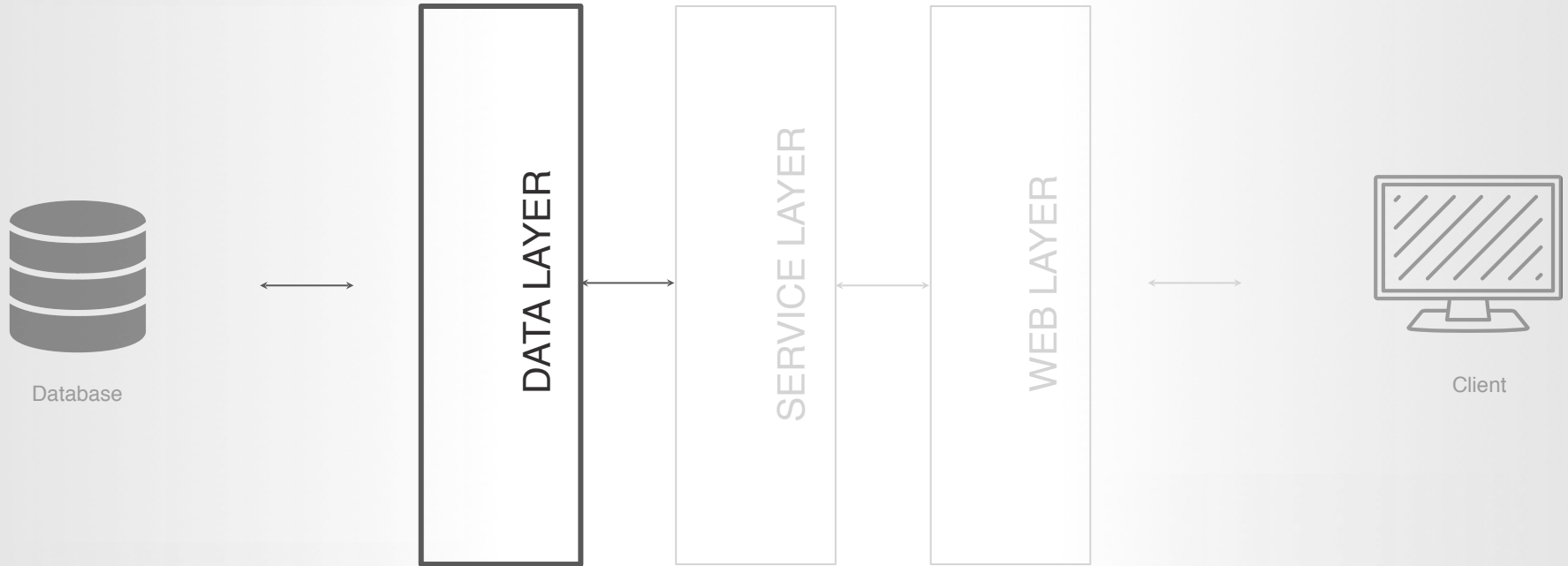
Server Structure

The service layer has a method that was called by the web layer. It's first job is to call a method from the data layer that gets all of Lucy's previous ratings.



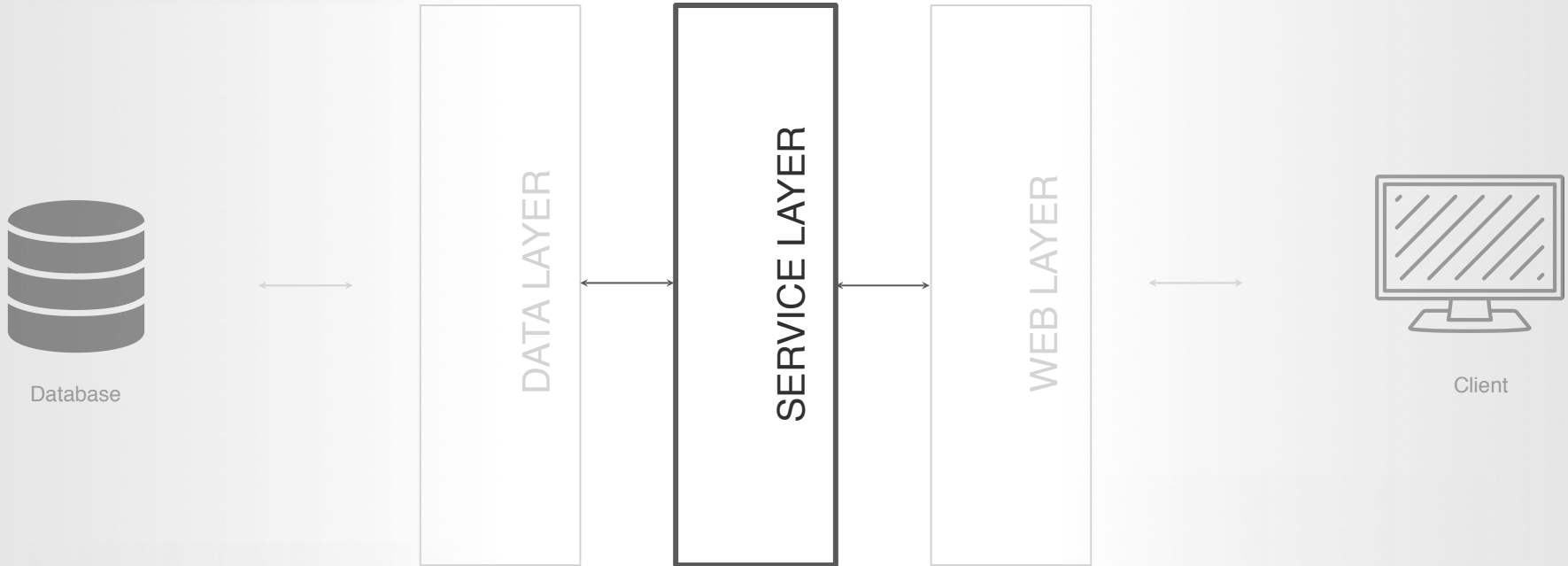
Server Structure

The data layer has a method that was called by the service layer. It's job is to ask the database for Lucy's data. It then returns this data to the service layer.



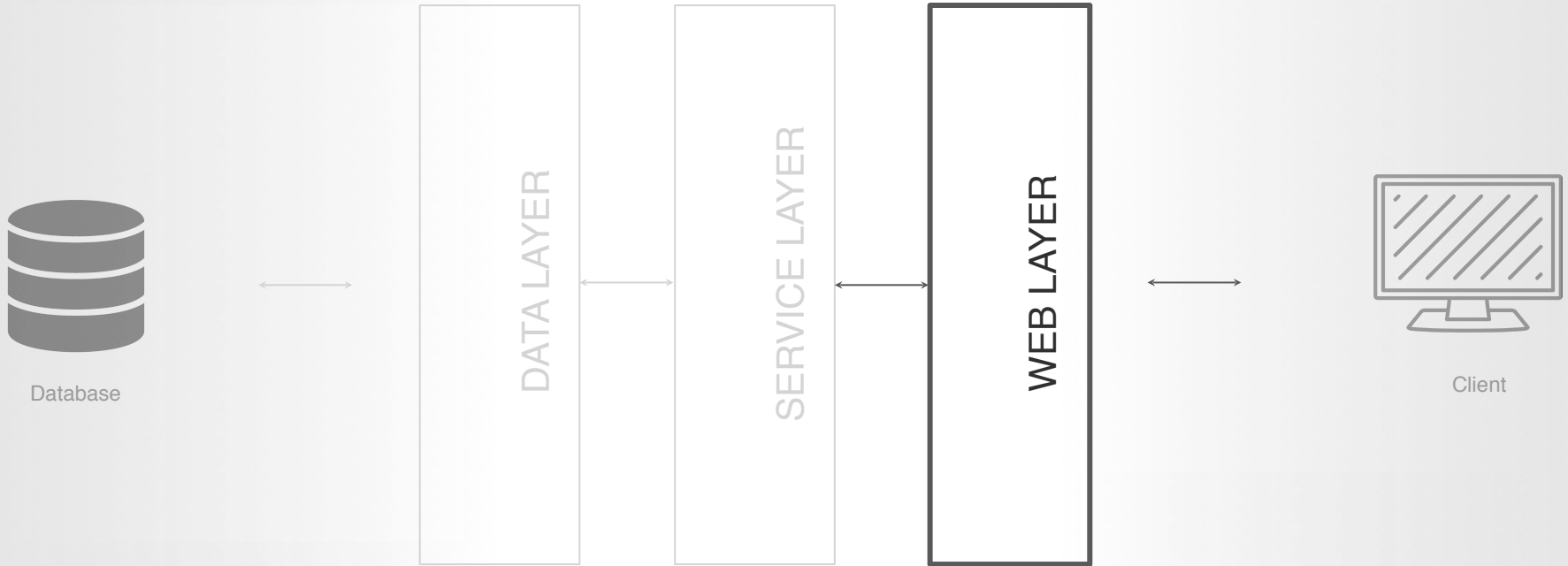
Server Structure

Now that the service layer has Lucy's data, it uses some complex algorithm to figure out which movie to recommend next. Once it knows the new movie list, it returns this list to the web layer. The service layer is kind of the brains of the operation.



Server Structure

The web layer now has the recommended movies and sends this list to Lucy's computer to render visually, so that Lucy can select a movie to watch.



OOP

OBJECT ORIENTED PROGRAMMING

What does all of this have to do with the classes and objects??

- Each layer is composed of 1 or more classes.
- The data layer would have a `Movie` class so that it could model the information that it received from the database as Java variables.
- The data layer would also have a `dataAccess` class with methods for getting movies from the database and adding movies to the database. The service layer would need to create a `dataAccess` object so that it can call these methods and get data as needed.
- The service layer has its own class(es). The web layer would need to create a new service object from this class to call the service methods like `getRecommendations`.



OOP

OBJECT ORIENTED PROGRAMMING

I'm lost!?!

- That's okay. This will all make sense in a couple of weeks. This is just a little introduction. Just keep what you can from this in the back of your mind.

Why is this so complex?

- Because applications are inherently complex. 3 Layers to your server may sound overly complex now, but imagine you had 20 different features on your web service each with its own database tables and unique constraints. It gets tricky quickly if you don't have the right infrastructure in place to handle it.





Check-in Time

- On a social media application in which layer would a class called Friends likely be found?
- What about a class with a method that receives client requests for upcoming events ?
- What about a class with methods for calculating events within a 50 mile radius?



lunch.

Constructors



Constructors

A METHOD FOR BUILDING NEW OBJECTS

Constructors are the method that is called by the new keyword plus the class Name (e.g. `new Student()`).

Constructors have a special syntax `public ClassName()`.

If you don't create a constructor, an empty default constructor is created for you.

You can overload constructors.



Constructors

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
public class Student {  
    private double gpa;  
    private String name;  
  
    public Student(String name, double gpa){  
        this.name = name;  
        this.gpa = gpa;  
    }  
}
```

Remember each object instantiated from the class gets its own copy of every property and method.

The `this` keyword refers to the object in which the code is currently running.



Constructors

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
public class Student {  
    private double gpa;  
    private String name;  
  
    public Student(String name, double gpa){  
        this.name = name;  
        this.gpa = gpa;  
    }  
}  
-----  
  
public static void main(String[] args){  
    Student arni = new Student("Arni", 3.7);  
    Student student2 = new Student("Beth", 3.95);  
}
```

Constructors



PAIR PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Create a new class called Dog
- Add three properties that all dogs should have
- Add getters and setters
- Create a constructor
- CHALLENGE: Overload the constructor to create 4 separate constructors*

* Use Google if you need help

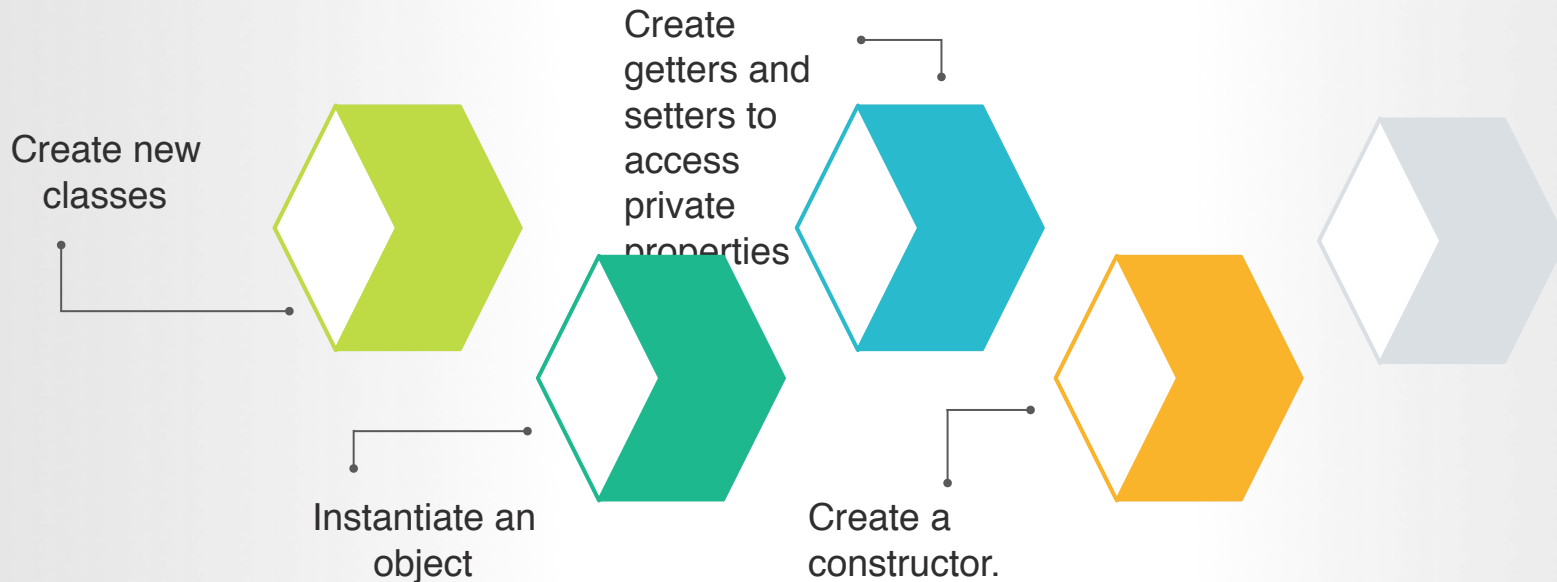


**10
minutes!**

Objectives & Key Outcomes

THE TAKEAWAYS FROM THIS CLASS

By the end of class today, you will be able to:





Stay Seated & Take 3 Deep Breaths.

RELAX.

Now take a short walk. Clear your head. After a few minutes break, quickly review your notes.
We'll start back in 10 minutes.



Solidifying Knowledge

INDEPENDENT PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work INDIVIDUALLY to complete all of the goals below.

Goals:

- Research Classes and Objects. Begin by Googling “Classes and Objects in Java”
- Read as many articles as possible as quickly as possible.
- Try watching a few short videos at 2x speed.
- If you feel very comfortable with classes and objects, try looking up getters, setters, constructors, and access modifiers in Java.

* Do NOT use this time to play games or visit social media sites. You have 3 months to change your life. Take full advantage of every minute.



**20
minutes!**

Solidifying Knowledge

PAIR PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Turn to your neighbor and take turns teaching each other everything you know about classes and objects.
- Look for gaps in your knowledge and research together to fill those gaps.



**20
minutes!**



Stay Seated & Take 3 Deep Breaths.

RELAX.

Now take a short walk. Clear your head. After a few minutes break, quickly review your notes.
We'll start back in 10 minutes.



Lab Time



LabTime

INDEPENDENT PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work together but INDEPENDENTLY write your own code to complete all of the goals below.

Goals:

- Complete all the katas listed in the activity file.

If this is tough, great! You're getting practice.

If this is easy, great! Help your buddies.

We'll be circulating to provide individual help where it's needed.



**50
minutes!**



Stay Seated & Take 3 Deep Breaths.


RELAX.

Now take a short walk. Clear your head. After a few minutes break, quickly review your notes.
We'll start back in 10 minutes.



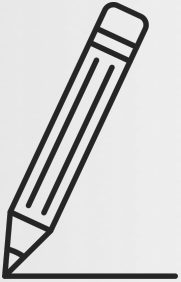
Sneak Peek

Note that next class will start with the sneak peek topic. You are NOT expected to master this today.



Inheritance

CLASS RELATIONSHIPS



Notebooks Ready? It's time for a mini lecture.



Inheritance

CLASS RELATIONSHIPS

Classes can inherit from each other. This is to say that they can have an is-a relationship.

A Poodle is a Dog, so the Poodle class should extend the Dog class.

extends is the key word used to indicate this relationship. When one class extends another, it inherits all the properties and methods of that class.



Constructors

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

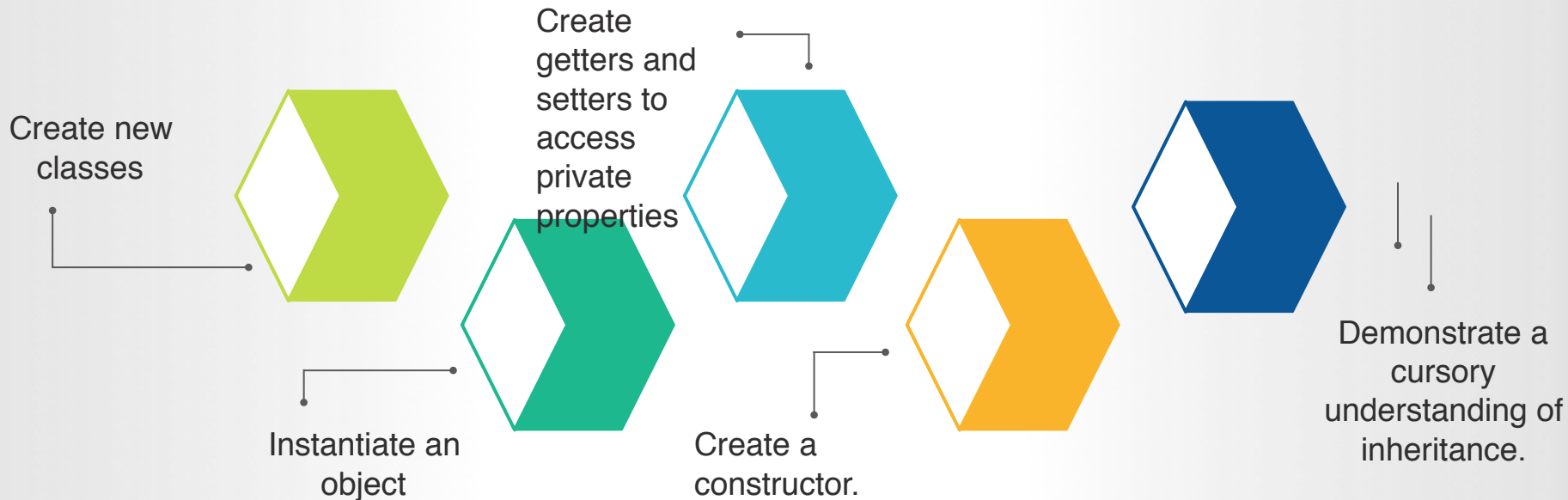
```
public class Dog {
    private String name;
    private int numLegs = 4;

    //getters and setter not shown
}
-----
public class Poodle extends Dog {
    private String coat = "curly hair";
}
-----
public static void main(String args[]){
    Poodle ralph = new Poodle();
    ralph.setName("Ralph the Poodliest of Poodles");
}
```

Objectives & Key Outcomes

THE TAKEAWAYS FROM THIS CLASS

By the end of class today, you will be able to:



Wrap Up

Module 2 Lesson 1

HOMEWORK

You don't have to submit your nightly homework, but you are expected to complete it.

For tonight's homework complete your katas

and watch the following video:

- <https://www.youtube.com/watch?v=MeP1CztNMdo>

and read the following articles:

- <https://www.guru99.com/java-oops-class-objects.html>
- <https://dzone.com/articles/understanding-classes-in-java-part-1>



Daily Assessment

You may leave after a staff member approves your assessment.



Daily Assessment

Work INDIVIDUALLY to complete all of the goals below.

Goals:

- Create a new project.
- Create a class called Book
- Add the following properties: title, author, isCheckedOut
- Add getters and setter for each property
- Add a constructor with 3 parameters
- Instantiate 2 Books (in other words, create 2 Book objects).