Welcome Aboard.
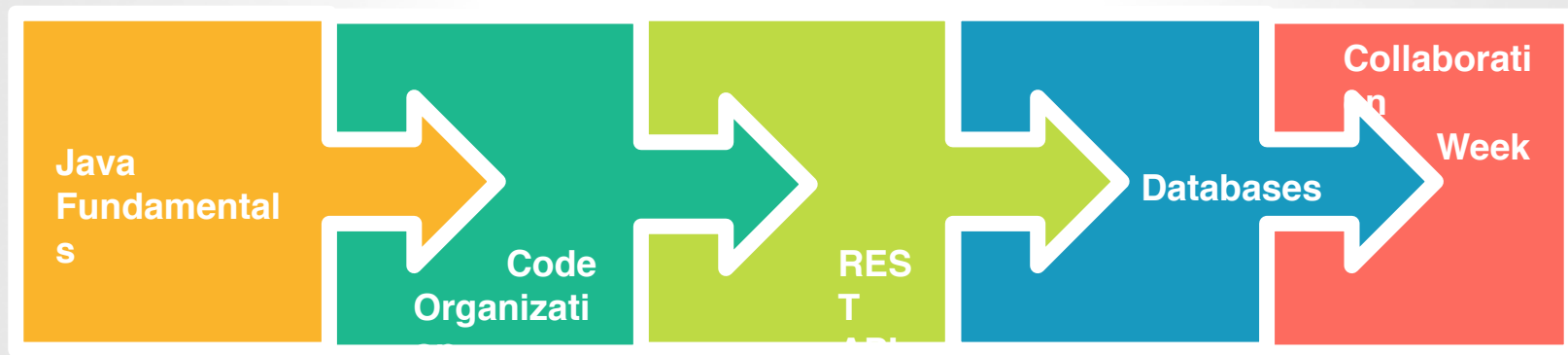
5 Units Of Java Content

There are 5 unique units of content each covering new set of skills followed by a capstone project.

Java Fundamentals

Code Organizati on

REST API

Databases

Collaborati on Week

KeyPoint

The first 3 days will be ROUGH. Do NOT fall behind. It gets easier, if you stay on top of it.

Objectives &Key Outcomes

THE TAKEAWAYS FROM THIS CLASS

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

Explain the component parts of a web application

Get user input.

Create variables and print their values.

Use operators to manipulate values.

Be familiar with using conditionals to determine which code executes.

Objectives &Key Outcomes

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

Let's Do This!

Anatomy of a Web App
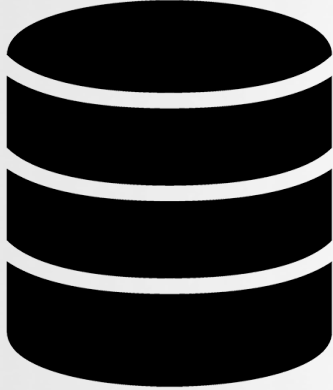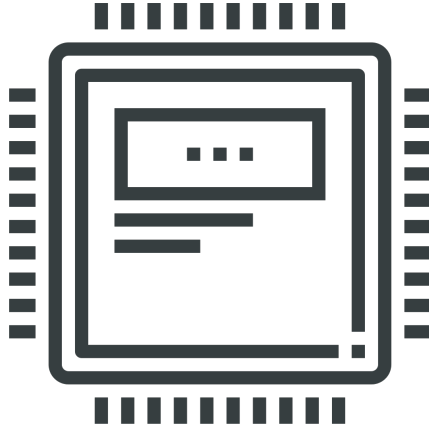
How the Web Works

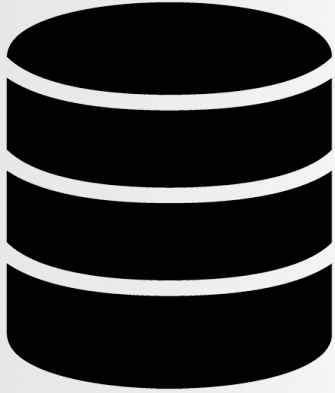Web applications are composed of 3 major components

Database

Server

Client

How the Web Works

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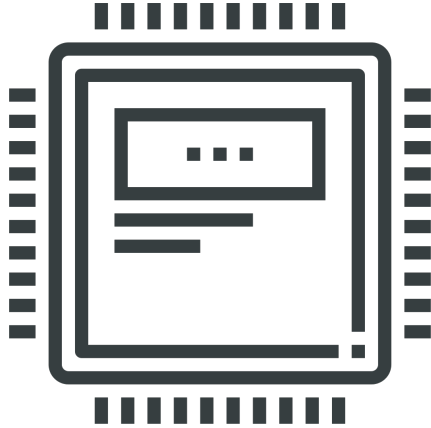
Database

Server

Client

How the Web Works

Servers 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

Check-in Time

- What are the 3 components of a web application?
- Which component stores data?
- Which renders the web page for the viewer?

How the Web Works

A simplified movie watching app. Let's call it Netflips. On Netflips, users can:

- Watch movies
- Watch the same movie repeatedly (because who doesn't watch Snakes on a Plane over and over?)
- Rate movies
- See a list of recommended movies

How the Web Works

Vickie frequents Netflips. All of her data will be stored and used to inform her recommended movie list.

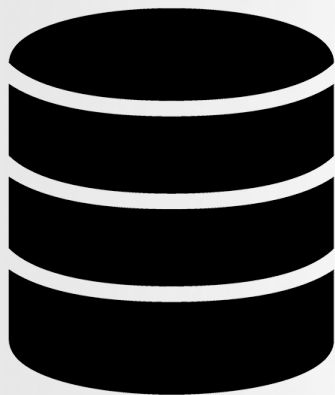Vickie binge watching all the all the Austin Powers movies.

Why, Vickie? whyyyyyy???!

How the Web Works

Databases are like complex spreadsheets.

Vickie's data

MOVIE	GENRE	REPEAT	RATIN
Old School	comedy	true	**G** 4.9
Jurassic Park	action	false	2.7
Bill and Ted	comedy	false	4.8
Mall Cop	comedy	true	5.0
It	horror	false	0.2

Database

How the Web Works

When Vickie goes to the recommended movie page, her computer asks the server which movies it should show in the list
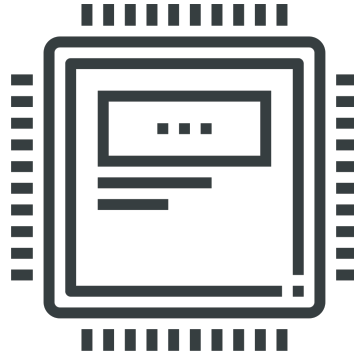
Server

Client

How the Web Works

The server requests Vickie's data from the database

Hey, Database! Can I see Vickie's past ratings?

Sure!
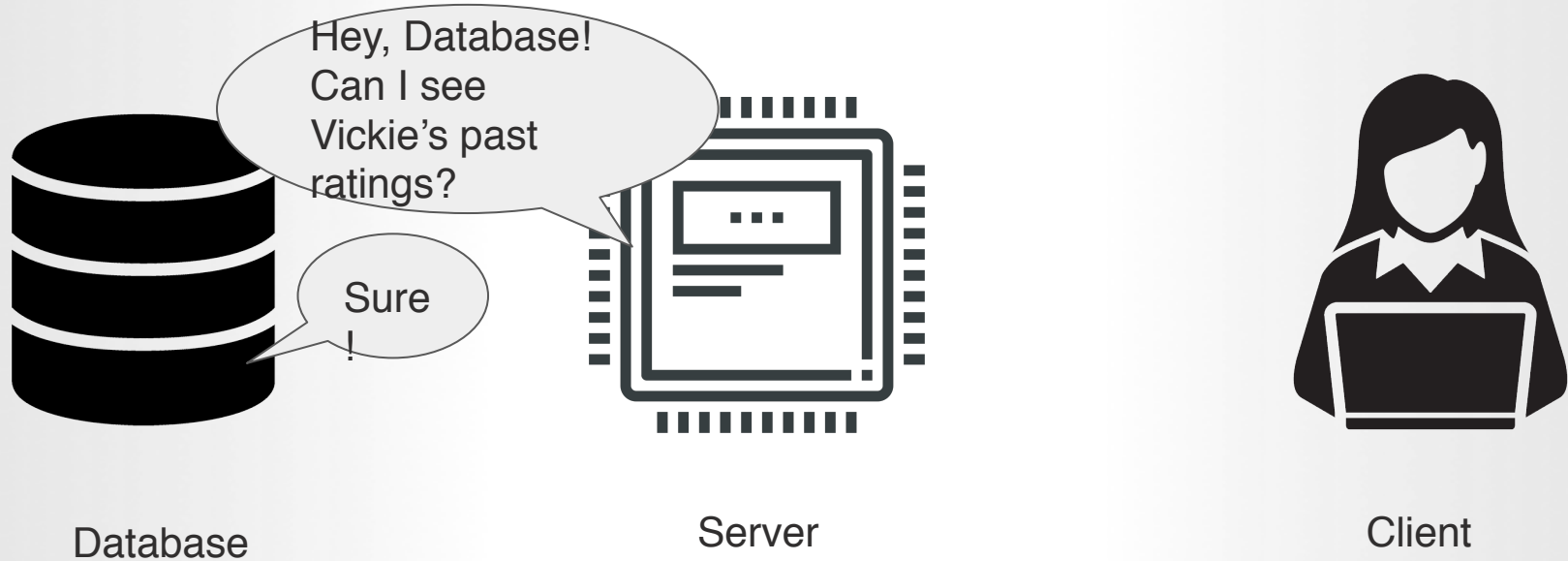
Database

Server

Client

How the Web Works

After the server determines what should be on the movie list, it sends it to Vickie's computer. Vickie's computer renders the data in the browser in an easy to read way.

You should watch Dumb and Dumber!
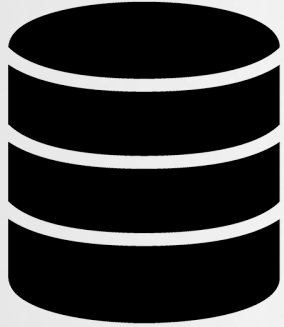
Database
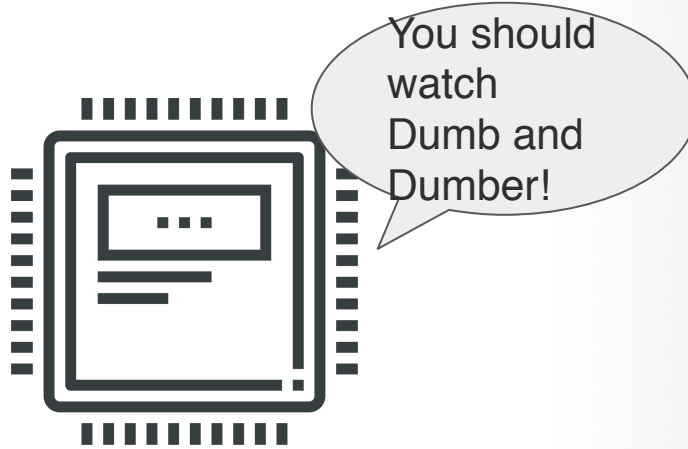
Server

Client

Check-in Time

- On a social media app, which component would handle rendering a list of friends in an attractive, easy to read, clickable way?
- Which component would calculate which friends' posts to display and in what order?
- Which component stores all your previous chat messages?

How the Web Works
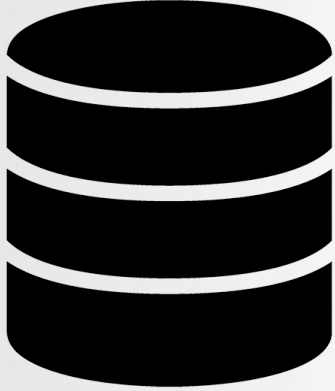
What language belongs to which component?

Database
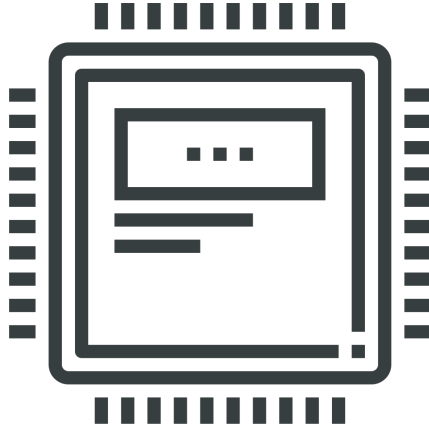
Server

Client

How the Web Works

What language belongs to which component?
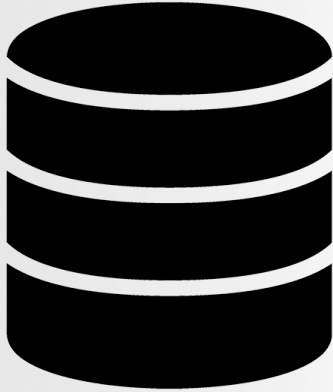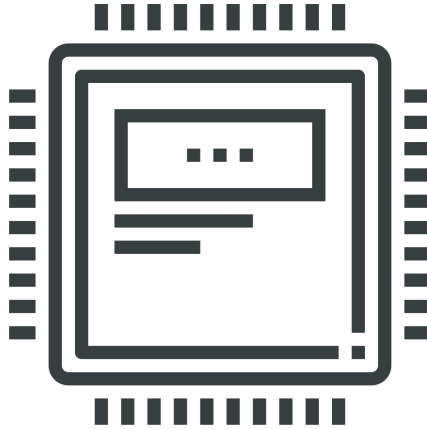
SQL	**Java**	**JavaScript, HTML, CSS**

Database	Server	Client

A Quick Aside

You don't need to write this down.

Is SQL the only database language?
- Nope. But it's the major one. Mongo is also worth learning.

Is Java the only server-side language?
- Nope. Not even close. C#, PHP, JavaScript, Python, Ruby, and a slew of other languages can be used. Java is incredibly powerful and popular though.

Is JavaScript the only client-side language?
- Yep. No getting around it. You must learn JavaScript to be a full-stack developer. JavaScript, HTML, and CSS have no major competitors.

A Quick Aside

Where are we starting?

We're starting with Java. Java is more structured than JavaScript and will help you to learn better coding practices.

Java is like learning to color in a coloring book. JavaScript is like coloring freehand. When you freehand, there are no mistakes because there are no rules. In a coloring book, you'll probably "mess up" more and get a little frustrated, but ultimately it teaches you better control which makes you better when you learn to freehand.

Learning Java is frustrating at first, but ultimately will help you learn better structure and patterns than learning in the rule-less world of JavaScript.

How the Web Works

Work in <u>PAIRS</u> to complete all of the goals below.

Goals:
- Take out a pen and paper and partner up with your neighbor
- Draw a quick diagram that explains how a web application works
- Explain to your neighbor all of the component parts of a web app
- Walk through your diagram using YouTube as an example site (What do you think is stored in the database? What logic is handled on the server? What happens when a user types "Dog on skateboard" in the search?)

5 minutes!

Objectives &Key Outcomes

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

Explain the component parts of a web application

Breathe

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 shortly.

Start Coding

Before We Begin

Let's quickly go over the slide color coding. It's pretty simple.
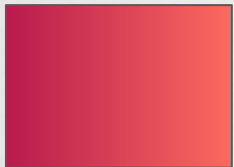
Red means stop. When you see a red header, <u>close your laptop</u>. All eyes should be on my screen. I need your full focus. I will not be answering questions that I just went over while you were busy on reddit.

Yellow means wait. Open your laptop and <u>code along with me</u>. Do not skip ahead even if you feel you already know this one. You'll be surprised how much you miss when you skip ahead. Stay focussed and type as I do.
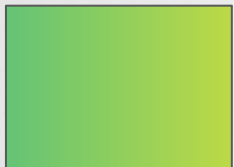
Green means go. This is independent practice <u>time so get to coding</u>! But remember, you're not alone. Once you've tried a few things, Google it. If you're still stuck after a few minutes of Googling, raise your hand and we'll come help.

Concept Title

WATCH & LEARN
Close your laptop. Eyes on my screen. Pay attention.

Sample
Watch & Learn

Concept Title

CODE-A-LONG
Open your laptop. Code with me. Don't jump ahead.

Sample Code-a-long

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

Sample Student Activity

Note the time here

10 minutes!

Introducing IntelliJ

Follow along with me as I complete this activity.

Goals:
- Create our first IntelliJ project
- Print "Hello World" to the console
- Create a second project
- Print "Goodnight" to the console

A Quick Aside

You don't need to write this down.

What is a class? What does main mean? Or public? Or String[]? What is all this garbage?
Don't worry about it. Or more specifically, don't worry about it today.

Lecture is boring. Coding is fun! I'm going to make an effort to break up all this background information over the next few days, so you don't have to sit through a 4 hour lecture. Trust me, we'll cover all of that in due time.

For now all you need to know is that all of your code should go inside the main method (that means inside the second set of curly braces).

Introducing IntelliJ

Work in <u>PAIRS</u> to complete all of the goals below.

Goals:
- Create a third IntelliJ project
- Print "Bananas are my favorite!"
- Create a fourth intelliJ project
- Ask your partner their favorite fruit
- Print "`[insert fruit here]` is `[insert partner's name here]` favorite!"

10 minutes!

Input & Variables

Input &Variables

Let's make this more interactive. Don't code along! Just watch. It'll be your turn in a minute.

- `Scanner scanner = new Scanner(System.in)` is a fancy way to say create a mechanism for retrieving user input.

- `String input = scanner.nextLine()` is a fancy way to say get the user input and call it "input".

- `System.out.println("You wrote " + input)` prints out the words "You wrote" plus whatever the user wrote in the console.

Input &Variables

Work in <u>PAIRS</u> to complete all of the goals below.

Goals:
- Using the provided example refactor your previous activity so that your partner can input their favorite fruit and the output will appear as below:
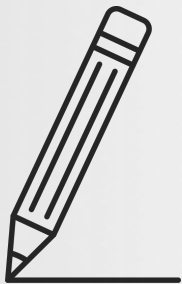- Print "`[insert fruit here]` is my partner's favorite!"

```
Scanner scanner = new Scanner(System.in);
String input = scanner.nextLine();
System.out.println("You wrote " + input);
```

10 minutes!

Variables &Types

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

Variables &Types

NUMBERS, BOOLEANS, & STRINGS

Programming is fundamentally about storing and manipulating values in an organized way.

Storing is the first step and this is where variables come into play.

Want to build a greeter application? The user inputs a name, you store it. You print "Hello " + the name you stored.

Want to build a simple adding machine? The user inputs a numerical value, you store it. The user inputs a second numerical value, you store it. You store a solution which is number 1 plus number 2. You print the solution.

Variables &Types

NUMBERS, BOOLEANS, & STRINGS

How do you store a value? 3 steps:

1. Tell the computer what <u>type of value</u> you want to store. The most essential types are
    a. Strings (words, sentences, etc like: "Woah there, buddy!")
    b. integers (whole numbers like: 1, 4, 100)
    c. doubles (decimals like: 1.25, 2.75, 3.14)
    d. boolean (true or false)

1. Next <u>name the variable</u>. This name can be almost anything you want. There are a few rules:
    a. Variable names must start with a letter, dash, underscore, or dollar sign. For now use letters.
    b. Variable names cannot have spaces
    c. Variable names should be camelCased

1. Assign the variable <u>a value using the equals sign</u>.

Variables &Types

What in the world are you talking about???

Okay.. so computers are dumb. If you input the number 4, your computer immediately forgets it, <u>unless</u> you explicitly tell it "hey, I'm going to give you an integer and I want to name it littleNumber".
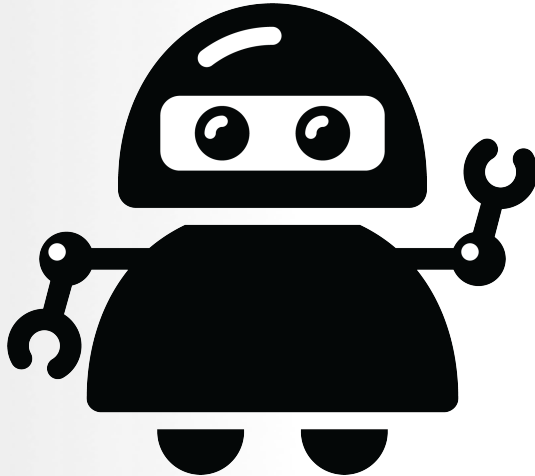
garbageBot
2.0

Variables & Types

```
String name = "Christopher Robin";

int age = 5;

double favNumber = 7.2456;

Boolean isWeird = true;
```

Check-in Time

- What's the difference between an integer and a double?
- What is a boolean?
- What three things are required to store a value?
- What's the keyword used to declare a variable that contains a series of alphanumeric characters (like a sentence)?

Input &Variables

This star indicates that this is a tough one!

Using the knowledge you've gained so far and the fact that `nextInt()` can be used in place of `nextLine()` from the last activity to get an integer input instead of a String input, work in <u>PAIRS</u> to complete all of the goals below.

Goals:
- Get an integer from the user.
- Get a second integer from the user.
- Print the sum of the 2 numbers.

* Try things. Break things. Use Google to help you. Get your hands dirty!

15 minutes!

Objectives &Key Outcomes

THE TAKEAWAYS FROM THIS CLASS

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

Explain the
component parts
of a web
application

Create variables
and print their
values.

Get user
input.

lunch.

Arithmetic Operators

Operators

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

Operators
ARITHMETIC OPERATORS

REMEMBER: Programming is fundamentally about storing and manipulating values in an organized way.

Let's talk about manipulating values.

We can combine numeric values using arithmetic operators:

+   -   *   /   %   ++   --   +=   -=   *=
                                                       /=

Arithmetic Operators

```
double age = 5.3;
double multiplier = 2;  //doubles can hold non-decimal numbers
int additionalNum = 3;
double product = age * multiplier;  //5.3 * 2 = 10.6

double sum = age + additionalNum;  //an int + a double is a double
age = 4;  //We can re-assign a variable a new value by writing variable name =
new num
age = age + 1;  //This is like writing age[new] = age[current] + 1;
age += 7;  //This is like writing age = age + 7;
age++;  //This is a short-hand for age = age + 1;
```

Arithmetic Operators

Work in <u>PAIRS</u> to complete all of the goals below.

Goals:
- Create a new project and paste the provided code in the main method. DO NOT RUN IT!
- With your partner, guess what each line will output
- Write down your guesses
- Run the file. Did your guesses match? If not, try to figure out what's actually happening.
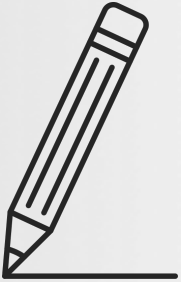
5 minutes!

Check-in Time

- What's does += mean?
- What is the difference between the pre and post-increment operators?
- What is the symbol for multiplication?
- What is the modulo (%) operator?

Operators

ARITHMETIC OPERATORS

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

Breathe

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 shortly.

Comparator Operators

Operators

ARITHMETIC & COMPARATORS

There are also comparator operators. Comparators are used to compare 2 or more values and create a Boolean value.

==   >   <   >=   <=   !=

Comparator Operators

```
int num1 = 4;
int num2 = 5;

Boolean isEqual = num1 == num2;  //single equals is the assignment
operator. Double equals is the comparator operator

Boolean isLarger = num1 > num2;
Boolean isSmaller = num1 < num2;
```

Logical Operators

Work in <u>PAIRS</u> to complete all of the goals below.

Goals:
- Create a new project
- Create an integer called magicNum and set it equal to 4
- Get an additional integer from the user using a Scanner
- Write a print statement that prints true if the user number is greater than or equal to magicNum and prints false otherwise.

10 minutes!

Objectives &Key Outcomes

THE TAKEAWAYS FROM THIS CLASS

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

Explain the component parts of a web application

Get user input.

Create variables and print their values.

Use operators to manipulate values.

Sneak Peek

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

Conditionals

FLOW CONTROL

Conditionals are a way to determine whether or not a block of code runs based on whether a given condition is true or false.

Logical Operators

```java
int num1 = 4;
int num2 = 5;

if(num1 == num2){
        System.out.println("These numbers are equal");
}else {
        System.out.println("These numbers are not equal");
}
```

Logical Operators

```
Scanner scanner = new Scanner(System.in);
String name = scanner.nextLine();

if(name.length() > 7){
        System.out.println("Long Name!");
}else {
        System.out.println("Meh. You're average.");
}
```

Objectives &Key Outcomes

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

Explain the component parts of a web application

Get user input.

Create variables and print their values.

Use operators to manipulate values.

Be familiar with using conditionals to determine which code executes.

Breathe

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 shortly.

Wrap Up

Module 1 Lesson 1

HOMEWORK

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

Read the article on Java numerical values: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html.

Complete the code katas.

Daily Assessment

You may leave after a staff member approves your assessment.

Daily Assessment

Work <u>INDIVIDUALLY</u> to complete all of the goals below.

Goals:
- Create a new project.
- Create a String variable called `name` that contains your name.
- Use a Scanner to get the users age as an integer.
- Create a boolean called `isOver21` that contains `true` if the age is over 21 and `false` otherwise.
- Print `[name here]is over twenty-one? [boolean value here].` substituting the values for `name` and `isOver21`, respectively.