Review OOP & UML

Presentation By Aria Shakoo

Instructor : Dr. Mojtaba Vahidi Asl

Spring 2025 AP

Table of contents

01

Why OOP

Why do we even need it?

02

Methods

Having Specific Functions

03

Constructors

What is it and what is it used for?

04

UML

Diagram

01

Why OOP?

OOP

- OOP can make software development more modular
- Which can make it easier to upgrade and update

- OOP systems can be easily upgraded from small to large systems
- It is very easy to partition the work in a project

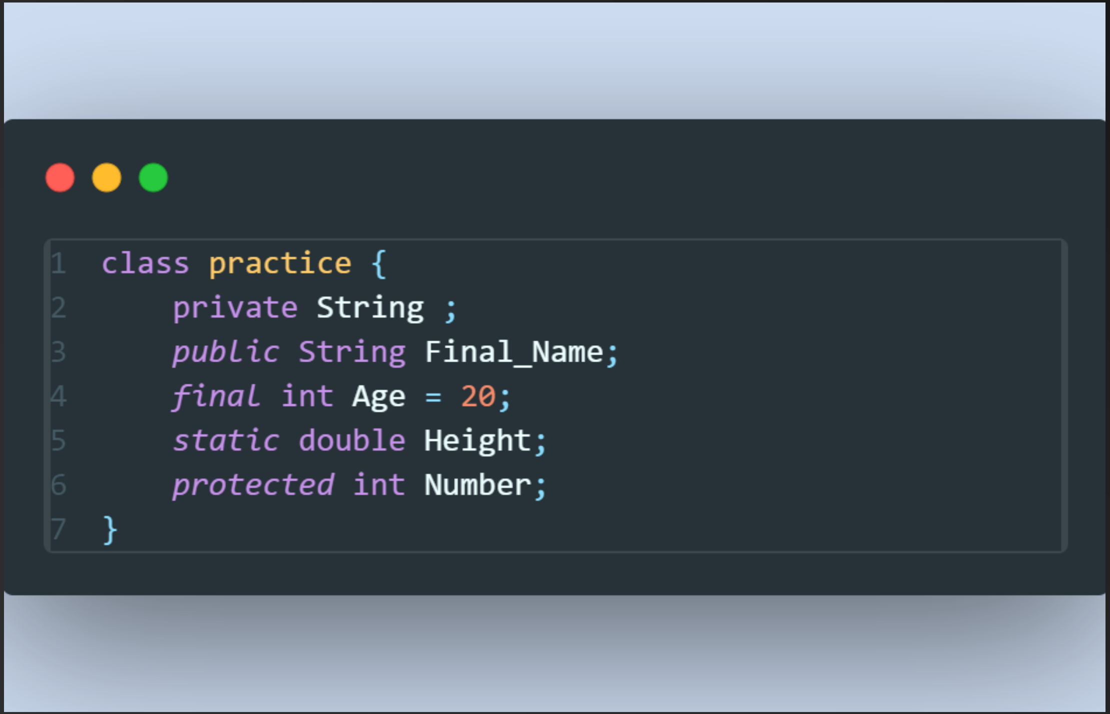
Class Attributes

- **Java class attributes** are the variables that are bound in a class

- A class attribute defines the state of the class during program execution

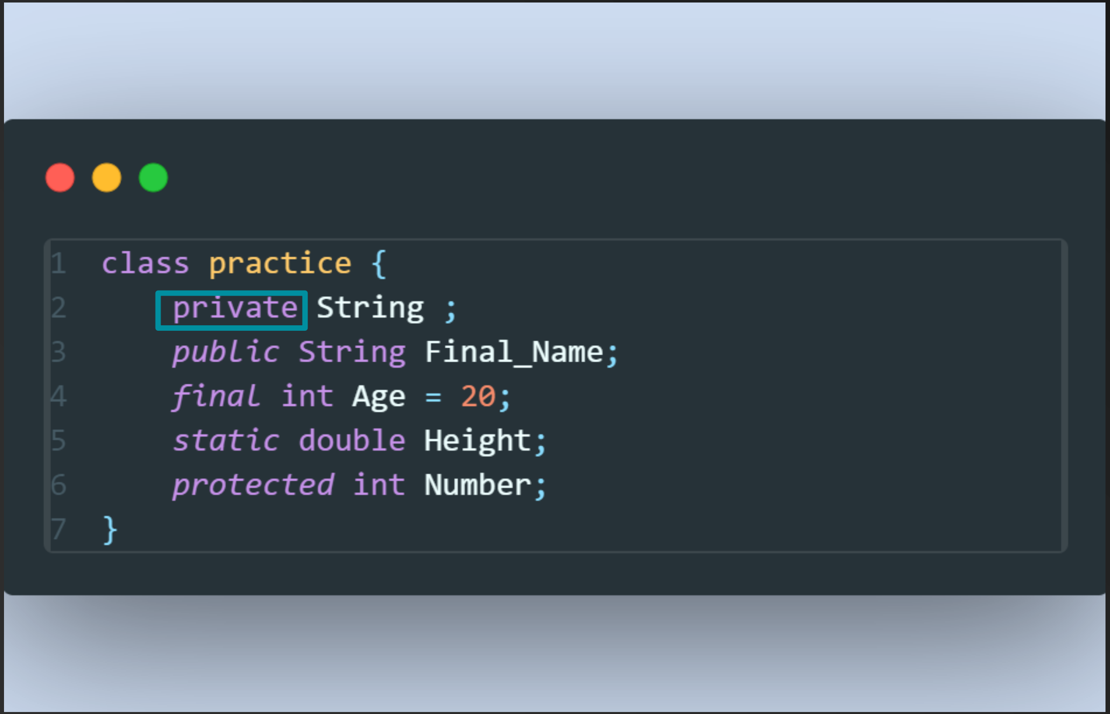
- A class attribute is accessible within class methods by default.

Lets See an Example!



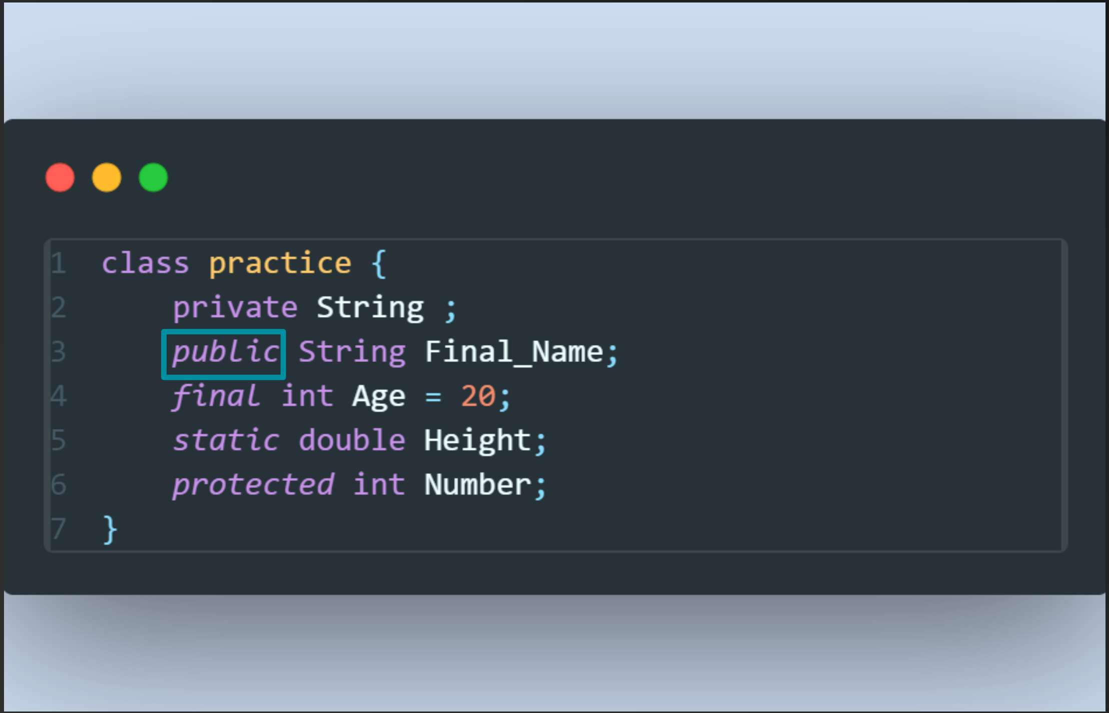
```
class practice {
    private String ;
    public String Final_Name;
    final int Age = 20;
    static double Height;
    protected int Number;
}
```

Lets See an Example!



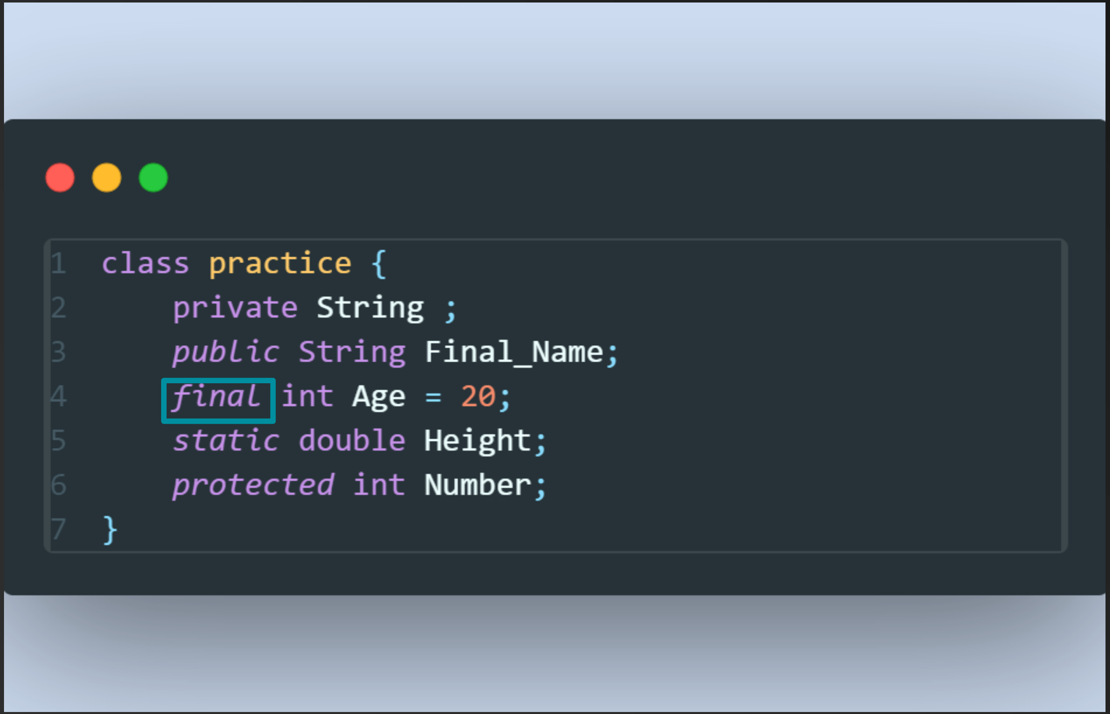
```
class practice {
    private String ;
    public String Final_Name;
    final int Age = 20;
    static double Height;
    protected int Number;
}
```

Lets See an Example!



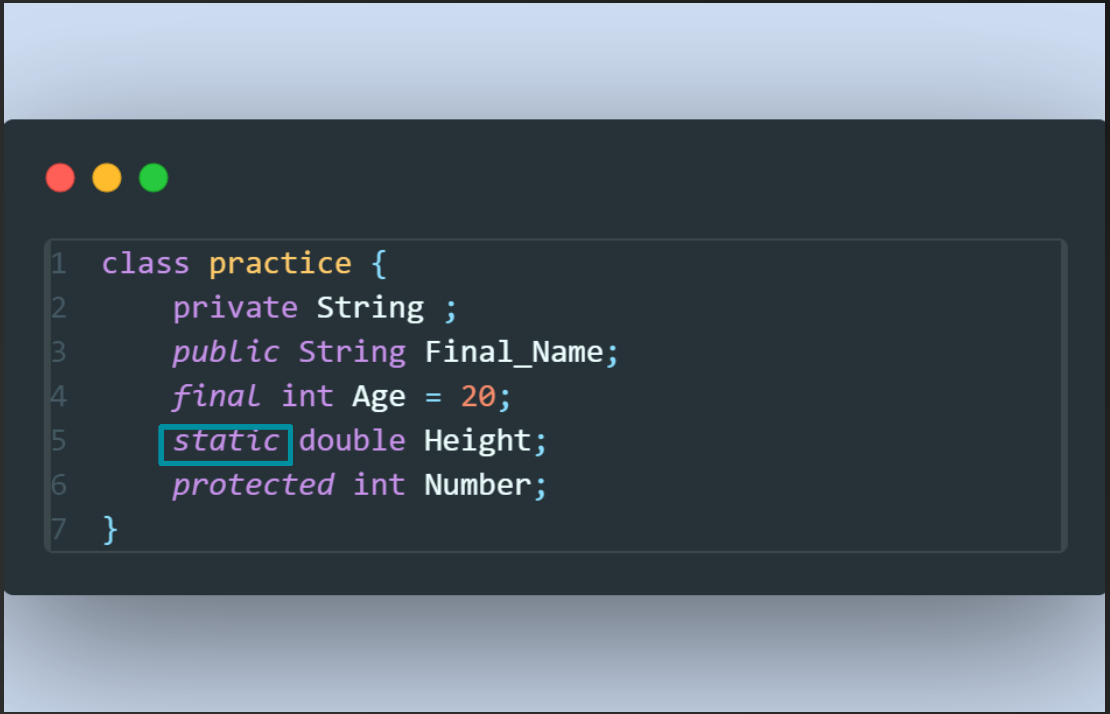
```
class practice {
    private String ;
    public String Final_Name;
    final int Age = 20;
    static double Height;
    protected int Number;
}
```

Lets See an Example!



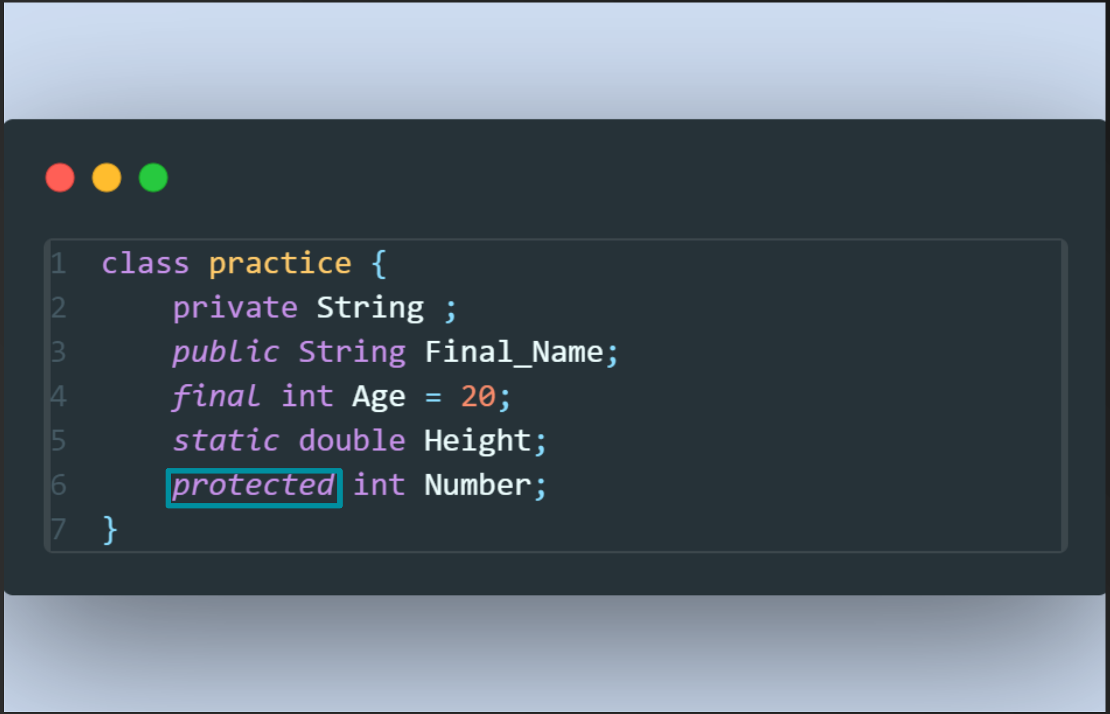
```
class practice {
    private String ;
    public String Final_Name;
    final int Age = 20;
    static double Height;
    protected int Number;
}
```

Lets See an Example!



```
class practice {
    private String ;
    public String Final_Name;
    final int Age = 20;
    static double Height;
    protected int Number;
}
```

Lets See an Example!



```
class practice {
    private String ;
    public String Final_Name;
    final int Age = 20;
    static double Height;
    protected int Number;
}
```

Access modifiers Review

Private

Protected

Public

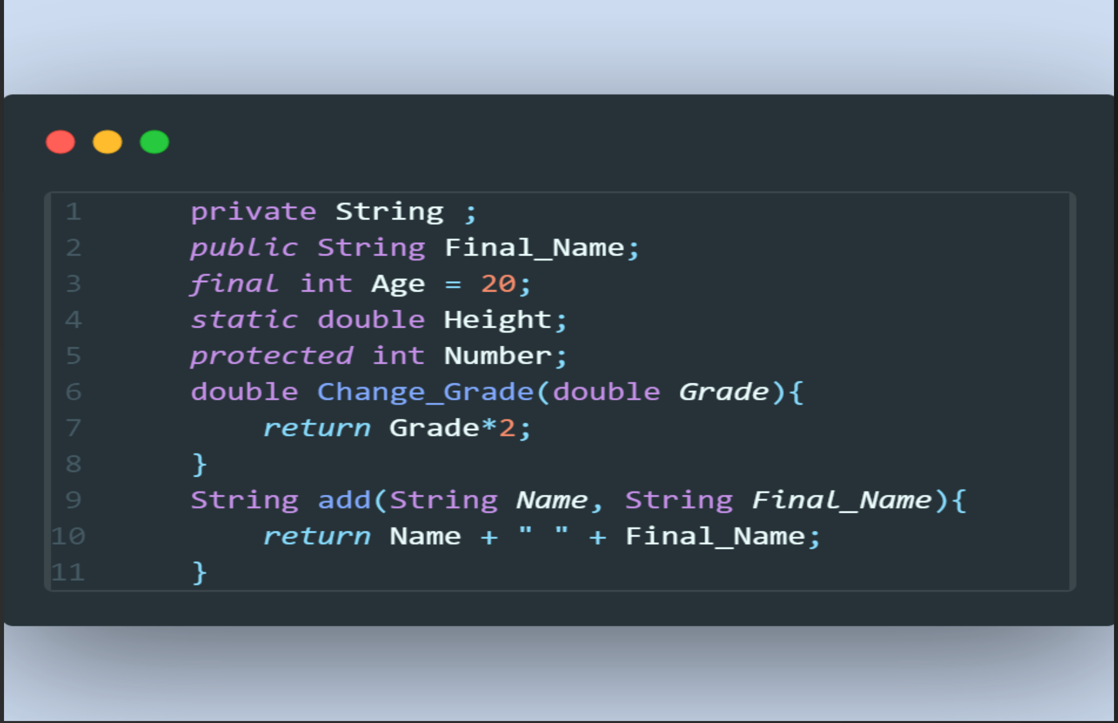
02

Methods

Class Methods

- A method in Java is a set of instructions that can be called for execution using the method name
- Getters and setters are used to protect your data

Lets See an Example!



```
private String ;
public String Final_Name;
final int Age = 20;
static double Height;
protected int Number;
double Change_Grade(double Grade){
    return Grade*2;
}
String add(String Name, String Final_Name){
    return Name + " " + Final_Name;
}
```

03

Constructors

Constructors

- A constructor is a special method in a class that initializes new objects.
- In most programming languages, the constructor has the same name as the class. (such as JAVA)

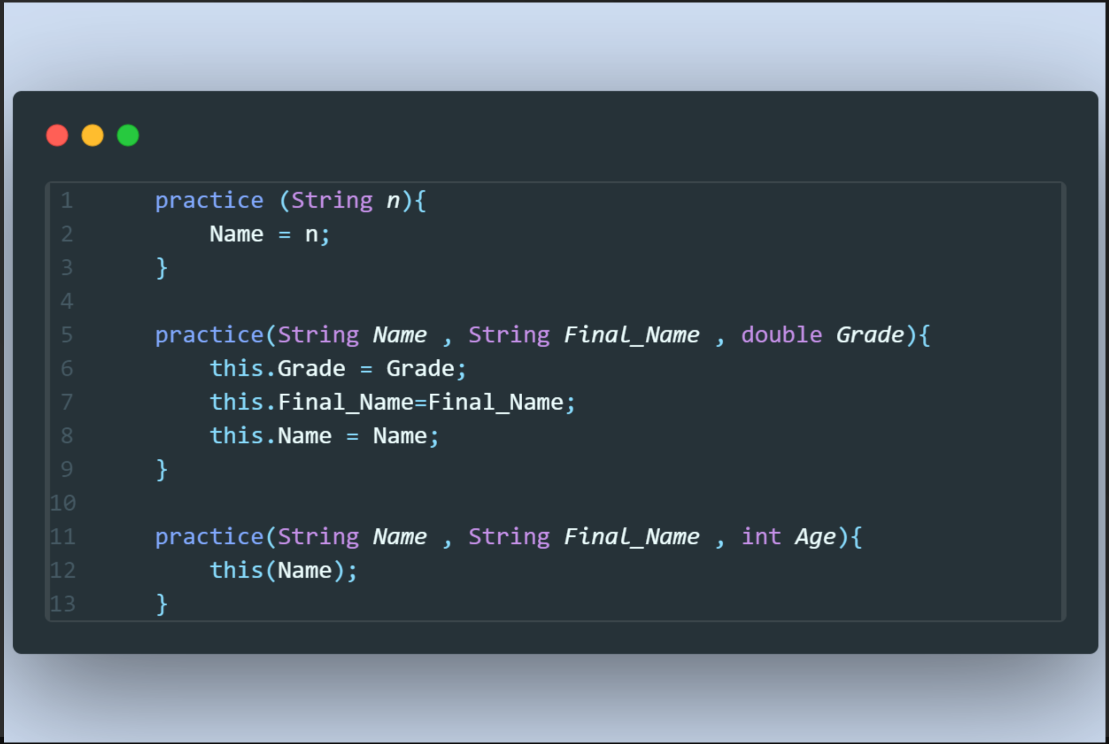
What are the types of Constructors?

No Arguements

Default

Parametrized

Lets See an Example!

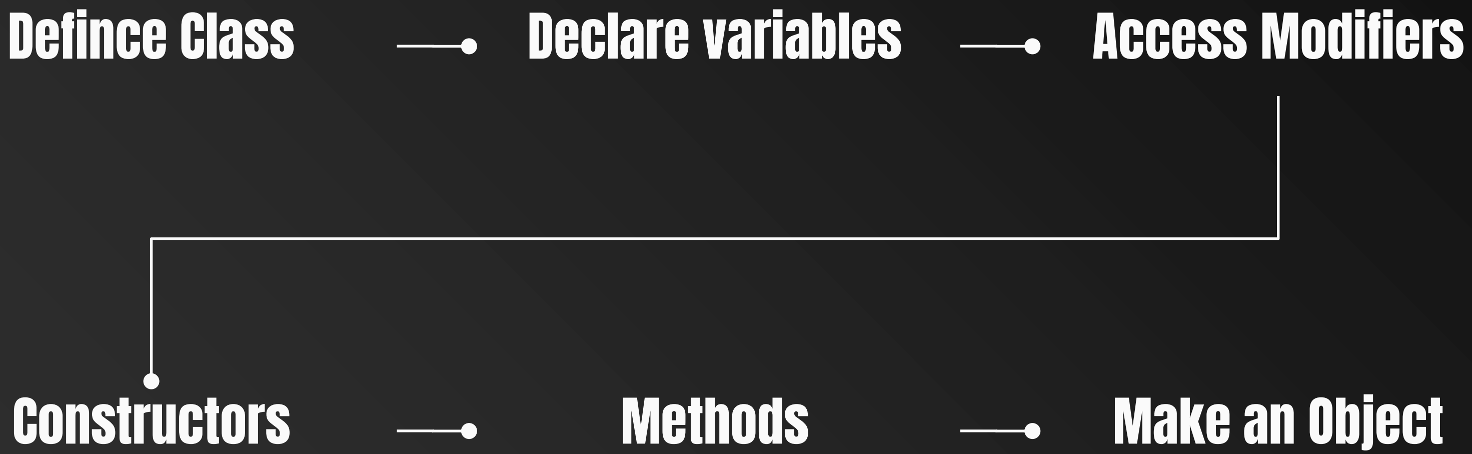


```
    practice (String n){
        Name = n;
    }

    practice(String Name , String Final_Name , double Grade){
        this.Grade = Grade;
        this.Final_Name=Final_Name;
        this.Name = Name;
    }

    practice(String Name , String Final_Name , int Age){
        this(Name);
    }
```

Sequence



Defince Class

Declare variables

Access Modifiers

Constructors

Methods

Make an Object

```
<div className="dropdown col-md-12">
  {/* level 1 */}
  <CSSTransition
  in={activeMenu === "main"}
  timeout={500}
  classNames="menu-primary"
  unmountOnExit
  onEnter={calcHeight}
  >
   <div className="menu">
    {categoryList.map((val) => {
     return (
      <DropdownItem
       leftIcon={val.icon}
       goToMenu={val.id}
       key={val.id}
       rightIcon={<RightArrowIcon />}
       >
       <h4>{val.name}</h4>
      </DropdownItem>
     );
    })}
    {linklv1List.map((val) => {
     return (
      <DropdownItem
       goToMenu="main"
       lefticon={<BoltIcon />}
       key={val.id}
       goHref={val.link}
       >
       {val.name}
      </DropdownItem>
     );
    })}
   </div>
  </CSSTransition>

  {/* level 2 */}
  <CSSTransition
   in={activeMenu === linkName}
   timeout={500}
   classNames="menu-secondary"
   unmountOnExit
   onEnter={calcHeight}
  >
   <div className="menu">
    <DropdownItem goToMenu="main" leftIcon={<ArrowIcon />}>
    <h4>Kembali</h4>
    </DropdownItem>

    {linkList.map((vallink) => {
     return (
      <DropdownItem
       lefticon={<BoltIcon />}
       goToMenu={linkName}
       goHref={vallink.link}
       >
       {vallink.name}
      </DropdownItem>
     );
    })}
   </div>
  </CSSTransition>
</div>v
```

```
"eslintConfig": {
  "extends": [
    "react-app",
    "react-app/"
  ]
},
```

```
{
position: fixed;
left: 0;
top: 0;
right: 0;
bottom: 0;

background-color: rgba(14, 7, 7, 0.514);
opacity: 0;
transition: all 0.3s ease-in-out;
pointer-events: none;

display: flex;
align-items: start;
justify-content: center;
}
},
"scripts": {
 "start": "react-scripts start",
 "build": "react-scripts build",
 "test": "react-scripts test",
 "eject": "react-scripts eject"
},
```

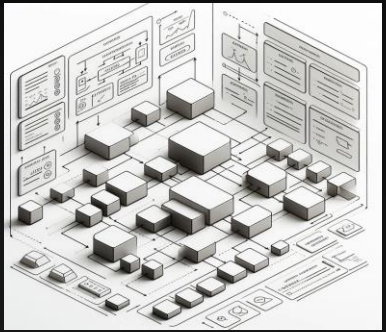
```
.App {
  text-align: center;
}
```

Uml Diagrams!

04

UML

UML

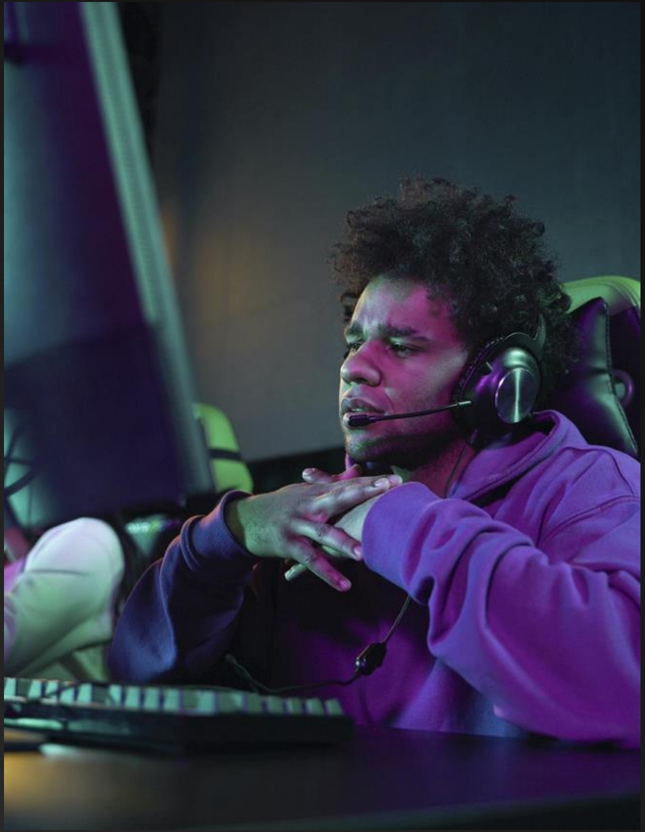


- The UML Class diagram is a graphical notation
- Is used to construct and visualize object oriented systems
- Is a type of static structure diagram
- Describes the structure of a system

Important Marks

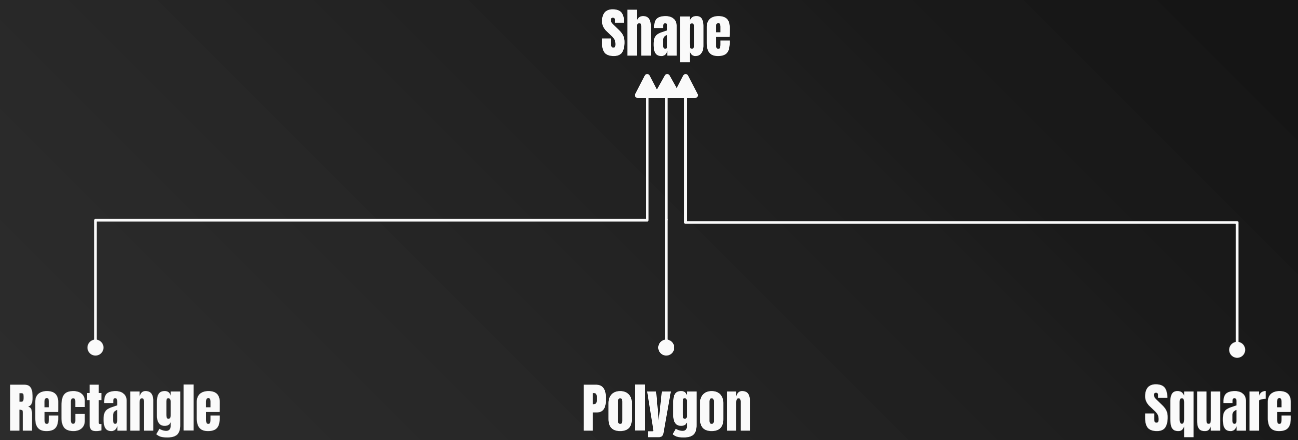
Public	Private	Protected
+	-	#
Package	Derived	Static
~	/	_

Why do we need UML Diagrams?



UML provides a standardized way to visualize the design of a system

Separate Targets



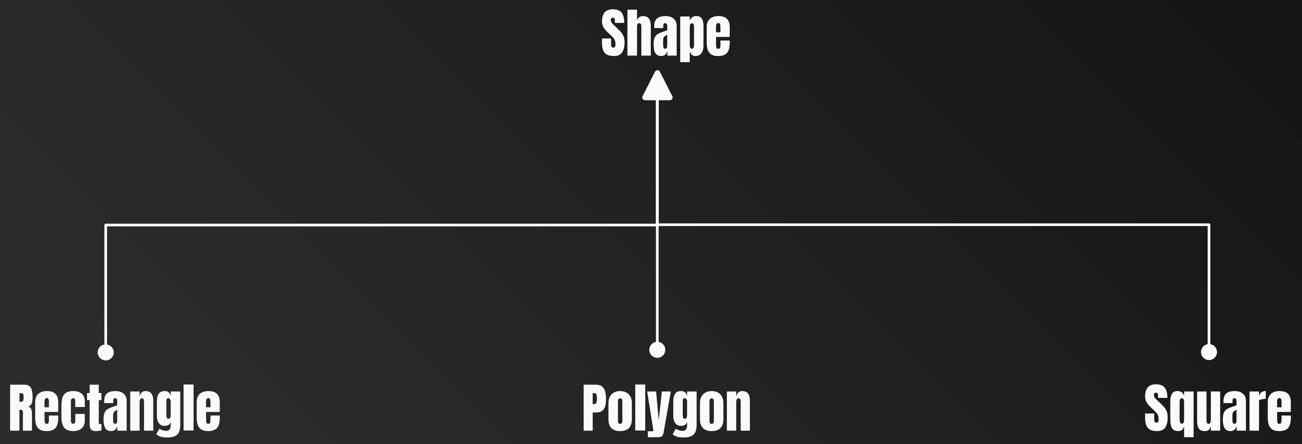
Shape

Rectangle

Polygon

Square

Shared Targets



Shape

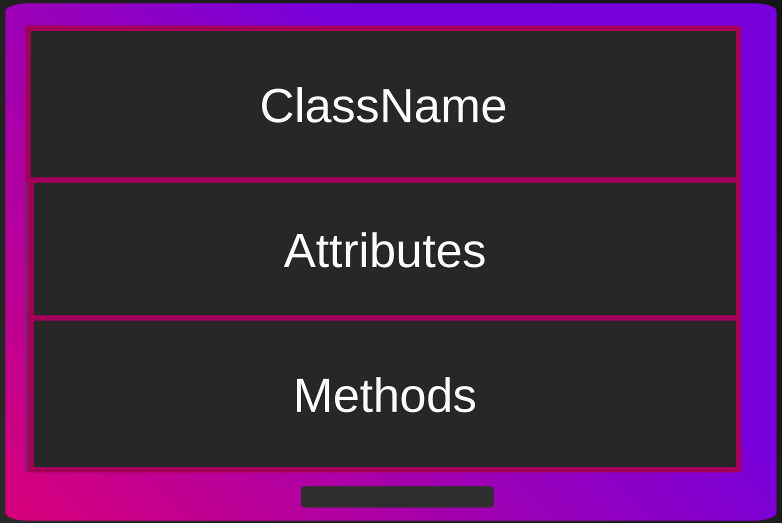
Rectangle

Polygon

Square

Info Shown

- Variable Names
- Access modifiers
- Methods and Their Arguements



ClassName

Attributes

Methods

Resources

https://www.w3schools.com/java/java_classes.asp
https://www.simplilearn.com/tutorials/java-tutorial/methods-in-java
https://www.javatpoint.com/java-constructor
https://www.programiz.com/java-programming/constructors

Any Questions?

Thanks!

s