



Review OOP & UML

Presentation By Aria Shakoo
Instructor : Dr. Mojtaba Vahidi Asl
1403 Ap Fall

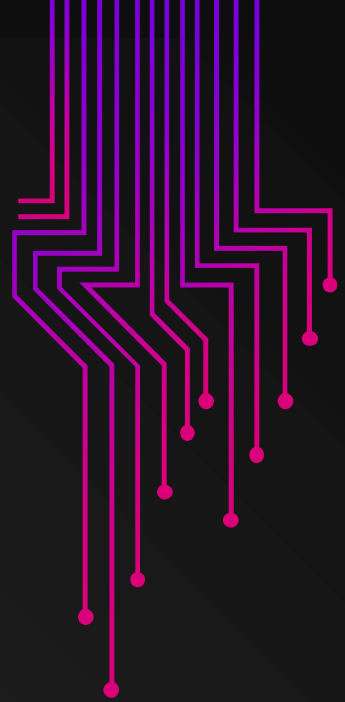
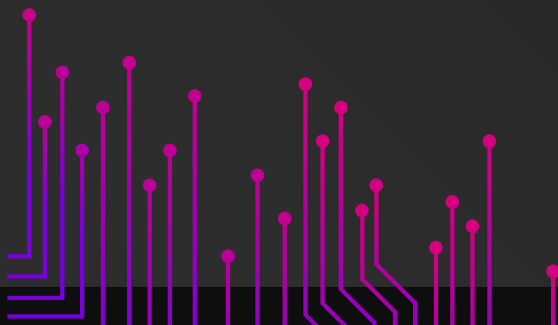


Table of contents

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?

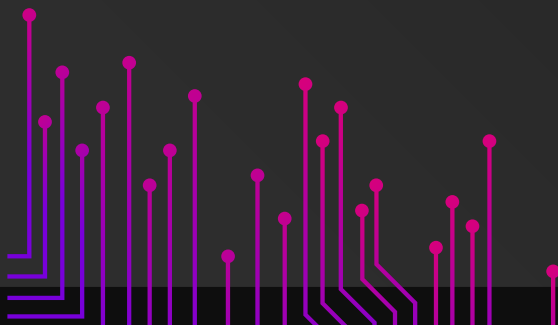
04

UML

Diagram

01

Why OOP?



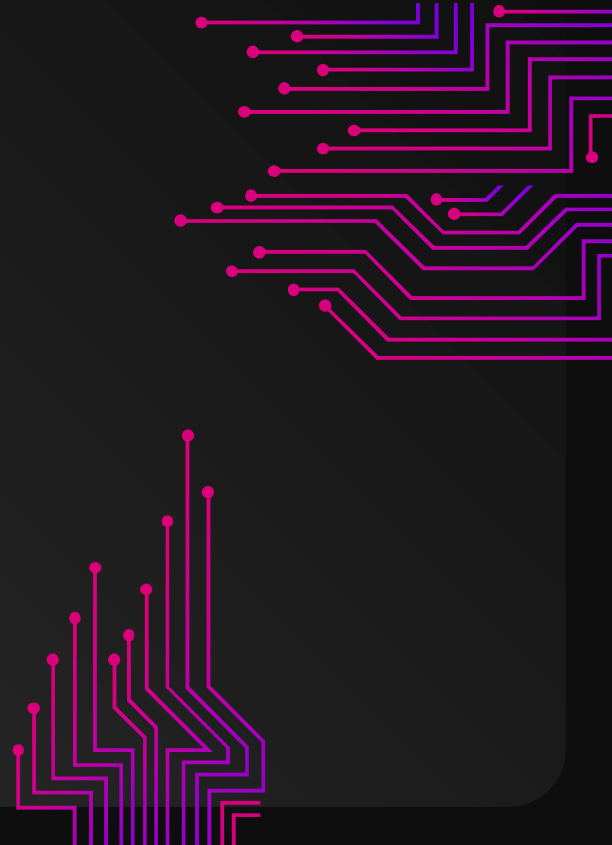
OOP



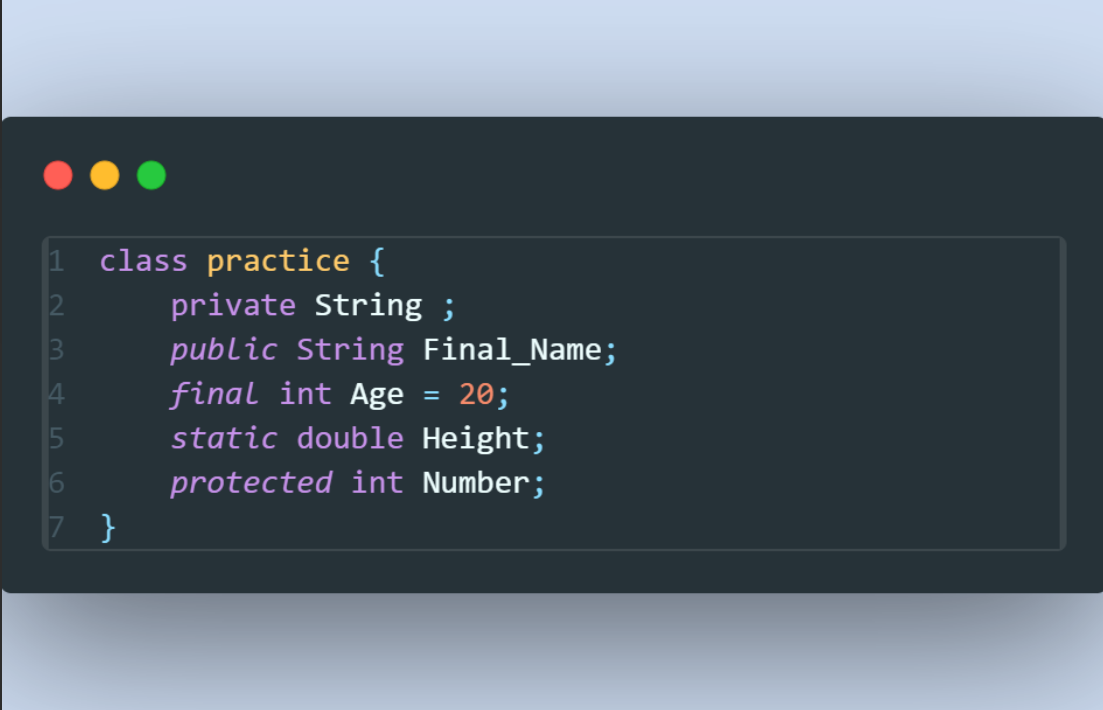
- OOP can make software development more modular
- Which can make it easier to upgrade and update
- OOP systems can be easily upgraded from small to large systems
- It is very easy to partition the work in a project

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!



```
1  class practice {  
2      private String ;  
3      public String Final_Name;  
4      final int Age = 20;  
5      static double Height;  
6      protected int Number;  
7  }
```

Lets See an Example!

```
1  class practice {  
2      private String ;  
3      public String Final_Name;  
4      final int Age = 20;  
5      static double Height;  
6      protected int Number;  
7  }
```

Lets See an Example!

```
1  class practice {  
2      private String ;  
3      public String Final_Name;  
4      final int Age = 20;  
5      static double Height;  
6      protected int Number;  
7  }
```


Lets See an Example!

```
1  class practice {  
2      private String ;  
3      public String Final_Name;  
4      final int Age = 20;  
5      static double Height;  
6      protected int Number;  
7  }
```

Lets See an Example!

```
1  class practice {  
2      private String ;  
3      public String Final_Name;  
4      final int Age = 20;  
5      static double Height;  
6      protected int Number;  
7  }
```

Lets See an Example!

```
1  class practice {  
2      private String ;  
3      public String Final_Name;  
4      final int Age = 20;  
5      static double Height;  
6      protected int Number;  
7  }
```

Access modifiers Review



Private



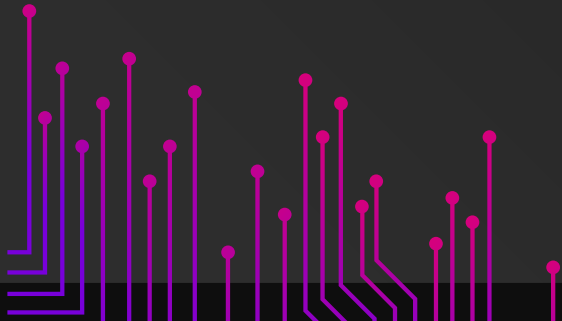
Protected



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

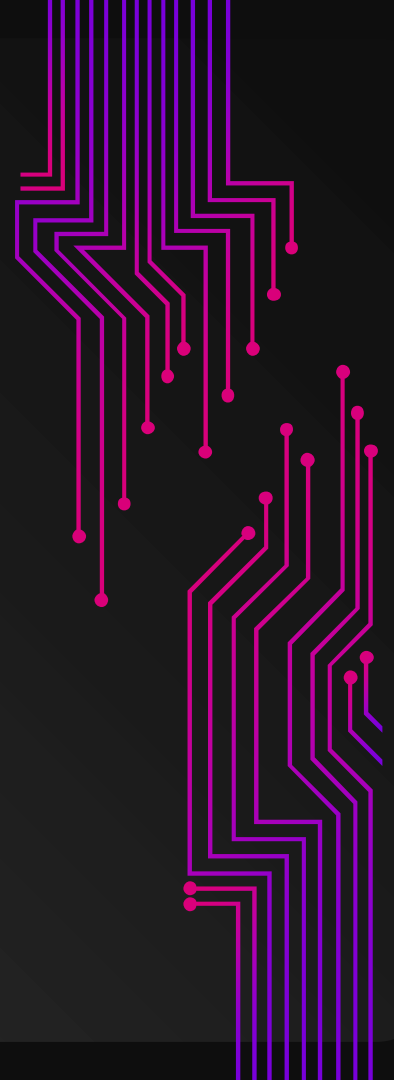
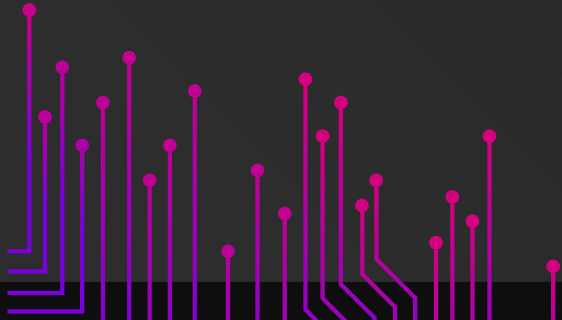
Lets See an Example!



```
1    private String ;  
2    public String Final_Name;  
3    final int Age = 20;  
4    static double Height;  
5    protected int Number;  
6    double Change_Grade(double Grade){  
7        return Grade*2;  
8    }  
9    String add(String Name, String Final_Name){  
10        return Name + " " + Final_Name;  
11    }
```

03

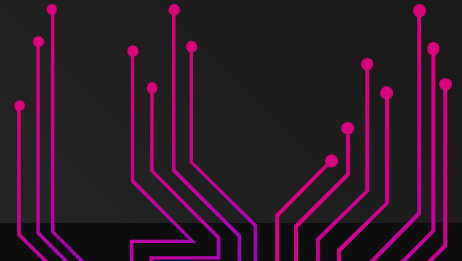
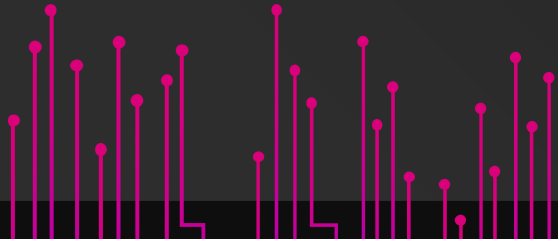
Constructors



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?



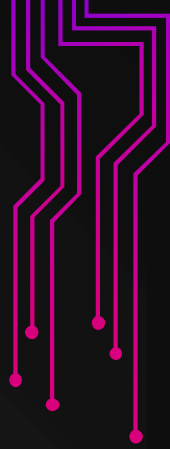
No Arguments



Default



Parametrized



Lets See an Example!

```
1  practice (String n){
2      Name = n;
3  }
4
5  practice(String Name , String Final_Name , double Grade){
6      this.Grade = Grade;
7      this.Final_Name=Final_Name;
8      this.Name = Name;
9  }
10
11 practice(String Name , String Final_Name , int Age){
12     this(Name);
13 }
```

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>
        );
      }
    </div>
    <BlinkList.map(val) => {
      return (
        <DropdownItem
          gotoMenu="main"
          leftIcon={<BoltIcon />}
          key={val.id}
          gotoRef={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>

    <BlinkList.map(val) => {
      return (
        <DropdownItem
          leftIcon={<BoltIcon />}
          gotoMenu={linkName}
          gotoRef={val.link}
        >
          {val.link.name}
        </DropdownItem>
      );
    }
  </div>
</CSSTransition>
</div>
```

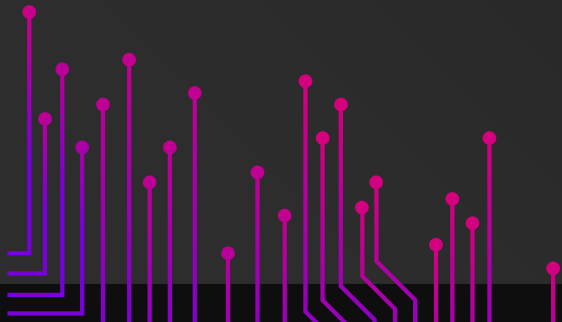
```
"eslintConfig": {
  "extends": [
    "react-app",
    "react-app/"
  ],
},
{
  position: fixed;
  left: 0;
  top: 0;
  right: 0;
  bottom: 0;
  background-color: rgba(14, 7, 7, 0.514);
  opacity: 0;
  transition: all 0.3s ease-in-out;
  pointer-events: none;
}
display: flex;
align-items: start;
justify-content: center;
<||>
```

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

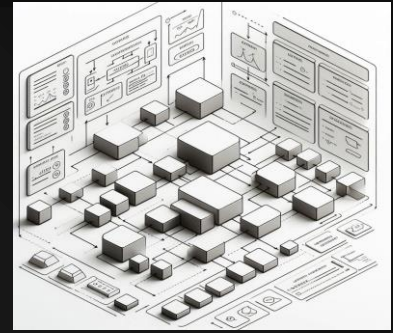
Uml Diagrams!

04

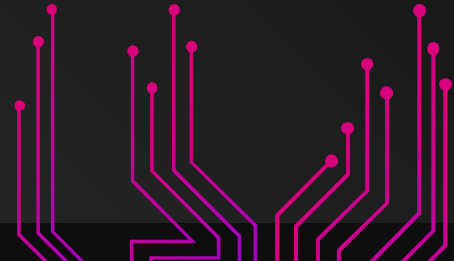
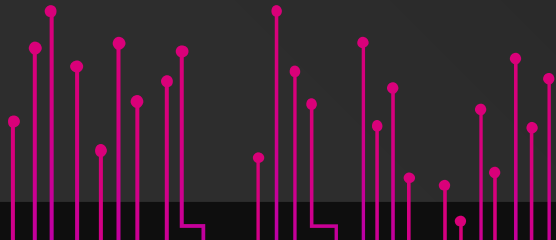
UML



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

+

Package

~

Private

-

Derived

/

Protected

#

Static

_

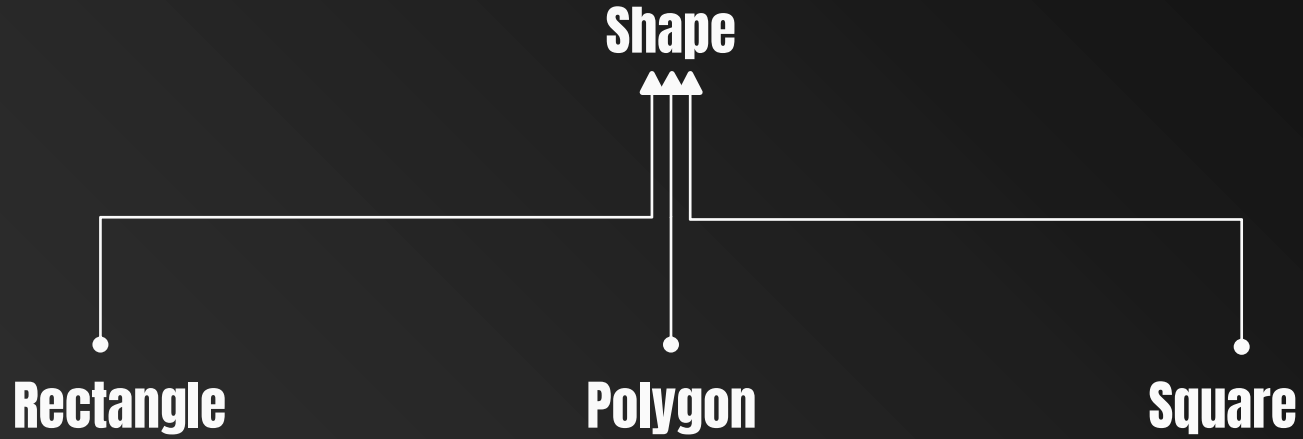
Why do we need UML Diagrams?



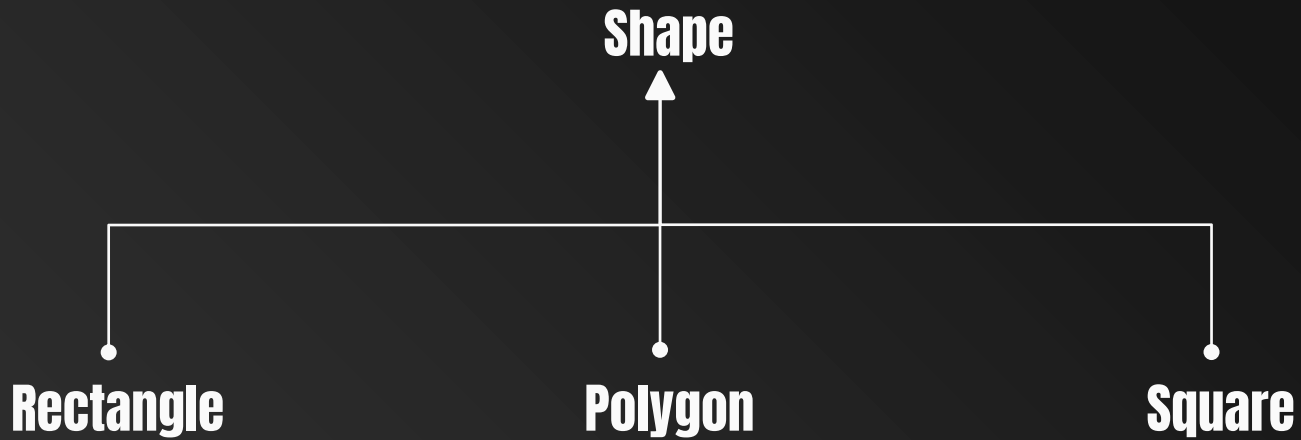
UML provides a standardized way to visualize the design of a system



Separate Targets



Shared Targets





Info Shown

- Variable Names
- Access modifiers
- Methods and Their Arguments



ClassName

Attributes

Methods



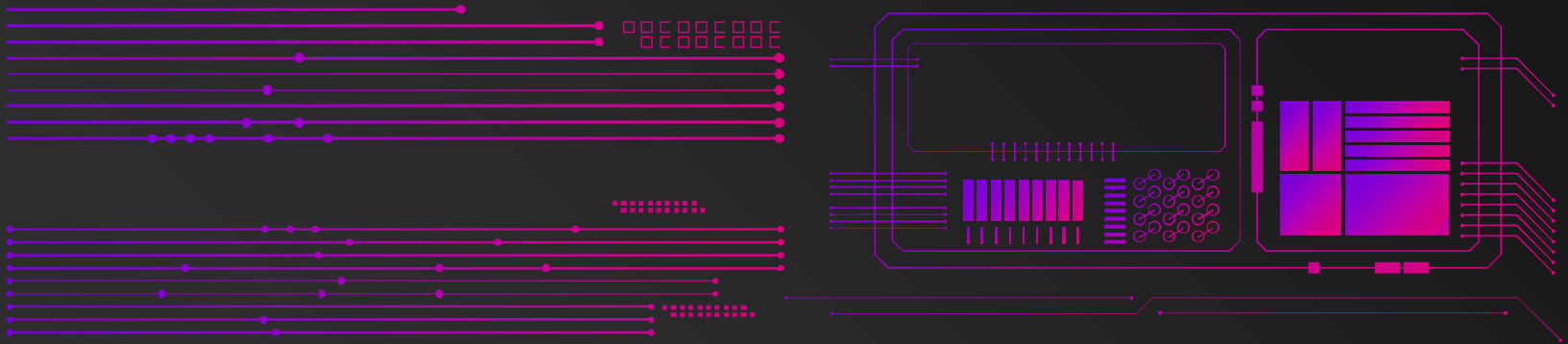
Resources

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!

