

# Java Track

```
study = filterByOrg ? study : filterByOrg == filterByOrg ? true  
status = filterByStatus ? study.status == filterByStatus : true  
matchStatus) {
```

```
return filterStudies() studies, filterByOrg = false, filterByStatus = true  
studies filter(study => {  
    return study.org == filterByOrg && study.status == filterByStatus
```

# OOP



## **Java Object Class**

[Java OOPs Concepts](#)

[Naming Convention](#)

[Object and Class](#)

[Method](#)

[Constructor](#)

[static keyword](#)

[this keyword](#)

## **Java Inheritance**

[Inheritance\(IS-A\)](#)

[Aggregation\(HAS-A\)](#)

## **Java Polymorphism**

[Method Overloading](#)

[Method Overriding](#)

[Covariant Return Type](#)

[super keyword](#)

[Instance Initializer block](#)

[final keyword](#)

[Runtime Polymorphism](#)

[Dynamic Binding](#)

[instance of operator](#)

```
copying = true  
status = true  
  
alterMyOrg = false, filter  
alterStudy => {  
    // ...  
}
```



## **Java Abstraction**

[Abstract class](#)

[Interface](#)

[Abstract vs Interface](#)

## **Java Encapsulation**

[Access Modifiers](#)

[Encapsulation](#)

Recommend Arabic videos resource:

<https://www.youtube.com/playlist?list=PLClnYL3l2AagY7fFlhCripLiIFybW3yQv> : [from #030 [JAVA] - Introduction to Object-Oriented Programming to #063 [JAVA] - Java Enum]

## Task: Library Management System

Design and implement a **simple Library Management System** using Java, focusing on object-oriented programming principles. The system should allow users to manage library resources, such as books, magazines, and DVDs.

**Requirements: next slide**

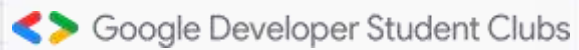


### 1. Define Object and Class & Methods & Constructor:

- Define classes for books, magazines, and DVDs with appropriate attributes (e.g., title, author, publication year).
- Implement methods to perform operations on library resources, such as adding, updating, and deleting items.
- Implement methods within classes to encapsulate behavior related to library resources (e.g., display information, check availability).
- Define constructors for classes to initialize object attributes during instantiation.

### 2. Implement the following concepts: Inheritance & Polymorphism & Abstraction & Encapsulation.

**3. Access Modifiers:** Use access modifiers (e.g., public, private, protected) to control access to class members based on visibility requirements



```
    status = filterByStatus ? study.status ==> filterByStatus : true  
    matchStatus) {
```

```
    return filterStudies() studies, filterByOrg = false, filterByStatus = true  
    studies filterStudy => {  
    }  
    return studies
```