

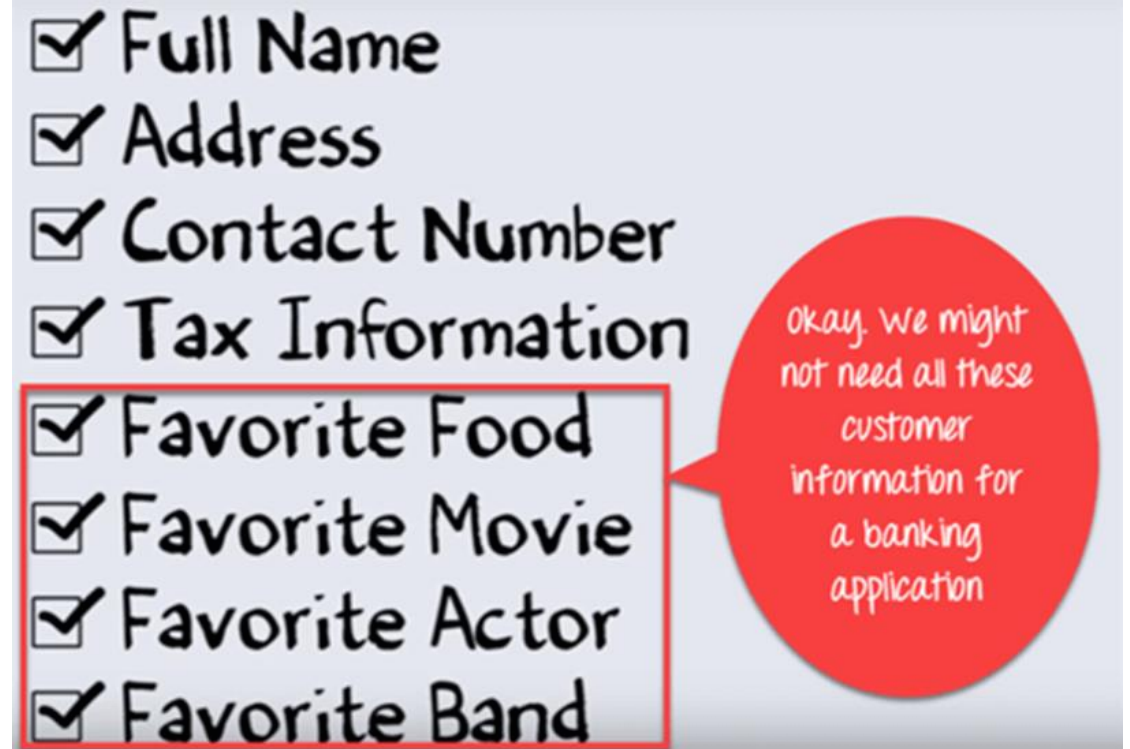
Abstraction in Java

What is Abstraction in OOPs? Learn with Java Example

- Selecting data from a larger pool to show only the relevant details to the object
- It helps to reduce programming complexity and effort.
- In Java, abstraction is accomplished using **Abstract classes** and **Interfaces**.
- It is one of the most important concepts of OOPs.

Abstraction - Example

- Suppose you want to create a banking application and you are asked to collect all the information about your customer
- There are chances that you will come up with following information about the customer
- But, not all of the above information to create a banking application



Abstraction - Example

- Select only the useful information for your banking
- Data like name, address, tax information, etc. make sense for a banking application



- ✓ Full Name
- ✓ Address
- ✓ Contact Number
- ✓ Tax Information

Abstraction - Example

- Since we have fetched/removed/selected the customer information from a larger pool, the process is referred as Abstraction.
- However, the same information once extracted can be used for a wide range of applications.
- For instance, you can use the same data for hospital application, job portal application, a Government database, etc. with little or no modification.
- Hence, it becomes your Master Data

Abstract classes

- It only indicates important things to the user and hides the internal details,
 - Eg: While sending SMS, you just type the text and send the message and you do not care about the internal processing of the message delivery

Abstract Class

- A class which is declared “abstract” is called as an abstract class.
- It can have abstract methods as well as concrete methods (well implemented methods).
- A normal class cannot have abstract methods.

Abstract methods

Abstract Method

- A method without a body is known as an Abstract Method
- It must be declared in an abstract class
- The abstract method will never be final because the abstract class must implement all the abstract methods

Rules of Abstract Method

- Abstract methods do not have an implementation; it only has method signature
- If a class is using an abstract method they must be declared abstract. The opposite cannot be true. This means that an abstract class does not necessarily have an abstract method.
- If a regular class extends an abstract class, then that class must implement all the abstract methods of the abstract parent

Difference between Abstraction and Encapsulation

Abstraction	Encapsulation
Abstraction solves the issues at the design level.	Encapsulation solves it implementation level.
Abstraction is about hiding unwanted details while showing most essential information.	Encapsulation means hiding the code and data into a single unit.
Abstraction allows focussing on what the information object must contain	Encapsulation means hiding the internal details or mechanics of how an object does something for security reasons.

Difference between Abstract Class and Interface

Abstract Class	Interface
An abstract class can have both abstract and non-abstract methods.	The interface can have only abstract methods.
It does not support multiple inheritances.	It supports multiple inheritances.
It can provide the implementation of the interface.	It can not provide the implementation of the abstract class.
An abstract class can have protected and abstract public methods.	An interface can have only have public abstract methods.
An abstract class can have final, static, or static final variable with any access specifier.	The interface can only have a public static final variable.

Advantages of Abstraction

- The main benefit of using an abstract class is that it allows you to group several related classes as siblings
- Abstraction helps to reduce the complexity of the design and implementation process of software

When to use Abstract Methods & Abstract Class?

- Abstract methods are mostly declared where two or more subclasses are also doing the same thing in different ways through different implementations
- It also extends the same Abstract class and offers different implementations of the abstract methods
- Abstract classes help to describe generic types of behaviors and object-oriented programming class hierarchy
- It also describes subclasses to offer implementation details of the abstract class