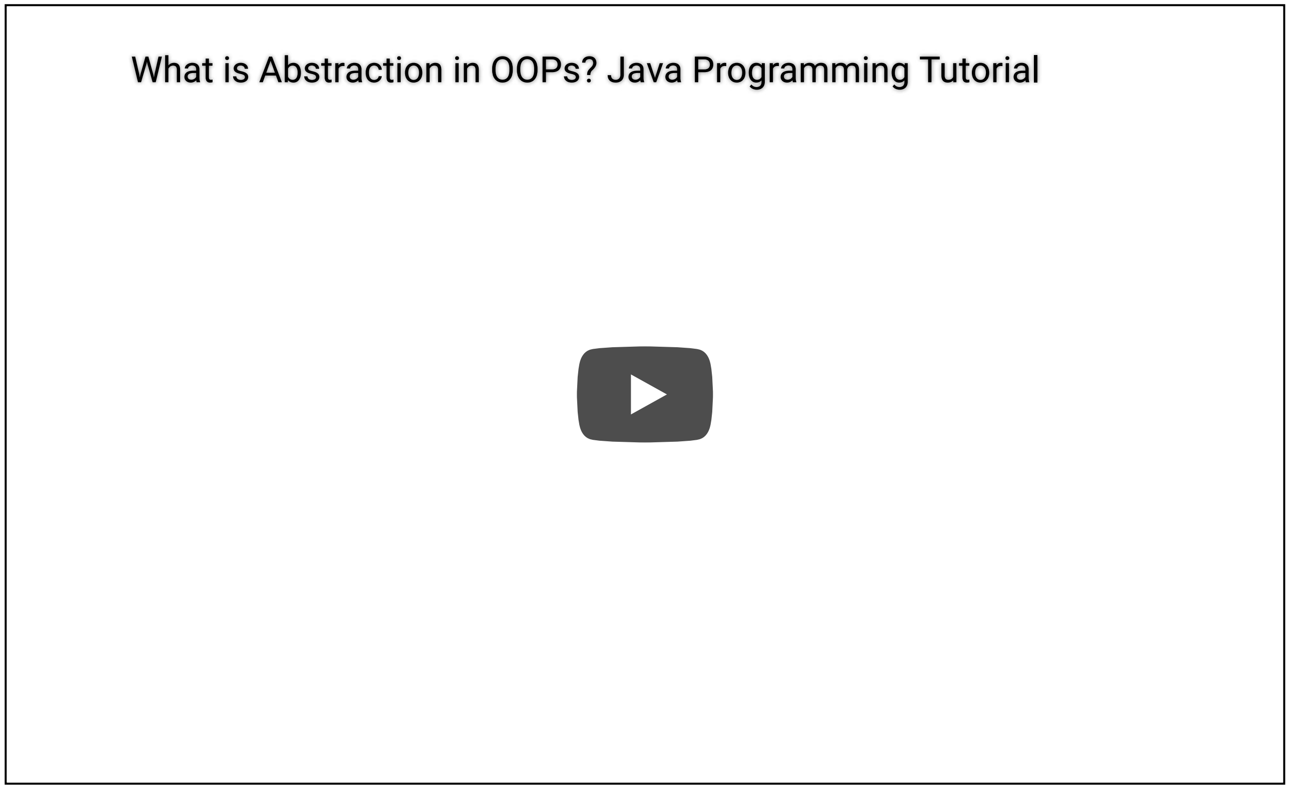
What is Abstraction in OOPs? Learn with Java Example

What is Abstraction in OOP?

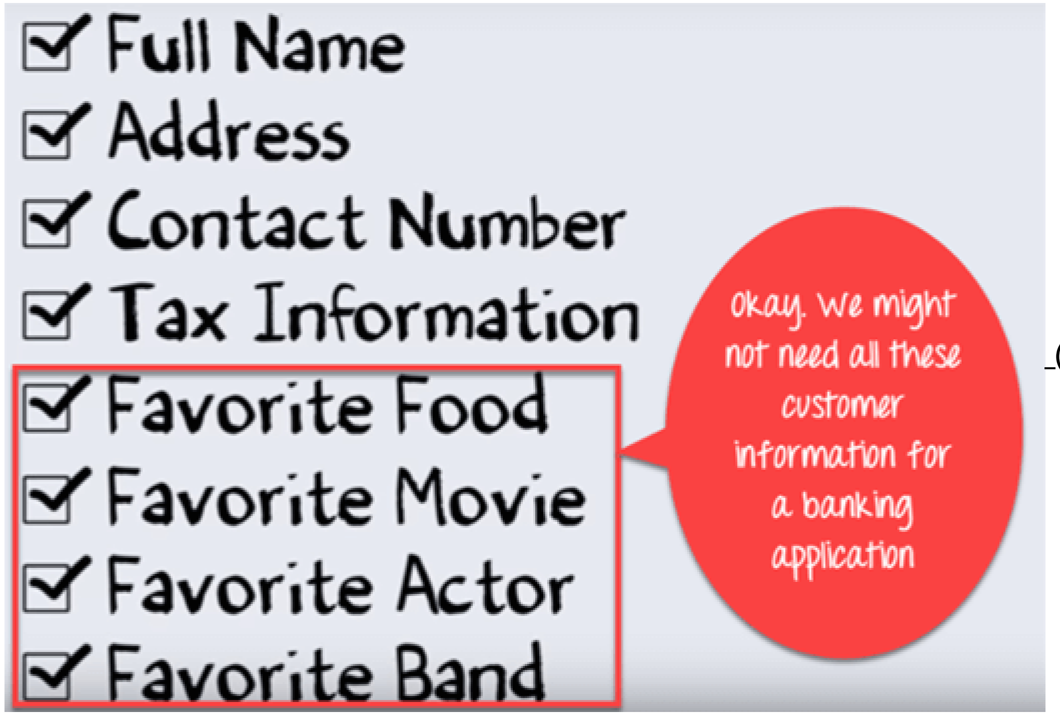
Abstraction is selecting data from a larger pool to show only the relevant details tc the object. It helps to reduce programming complexity and effort. In Java, abstraction is accomplished using Abstract classes and interfaces. It is one of the most important concepts of OOPs.



Please be patient. The Video will load in some time. If you still face issue viewing video click here (/faq.html#1)

Let’s Study Abstraction concept with an 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



(/images/abstraction_in_oops_112.png)

But, not all of the above information to create a banking application.

So, you need to select only the useful information for your banking application from that pool. Data like name, address, tax information, etc. make sense for a banking application

Full Name
Address
Contact Number
Tax Information

(/images/abstraction_in_oops_2.png)

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. This is an advantage of Abstraction.

How to achieve Abstraction?

At a higher level, Abstraction is a process of hiding the implementation details and showing only functionality to the user. It only indicates important things to the user and hides the internal details, ie. While sending SMS, you just type the text and send the message. Here, you do not care about the internal processing of the message delivery. Abstraction can be achieved using Abstract Class and Abstract Method in Java.

Abstract Class

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

Summary:

- Abstraction is the process of selecting important data sets for an Object in your software and leaving out the insignificant ones.
- Once you have modeled your object using Abstraction, the same set of data could be used in different applications.
- Java, abstraction is accomplished using Abstract classes and interfaces. We will study in detail about Abstract classes and interfaces in further tutorials.

Prev (/java-oops-concept.html) **Report a Bug** Next (/java-oops-encapsulation.html)

YOU MIGHT LIKE:

JAVA TUTORIALS

(/generate-random-number-java.html)
(/generate-random-number-java.html)

How to easily Generate Random Numbers in Java
(/generate-random-number-java.html)

JAVA TUTORIALS

(/multithreading-java.html)
(/multithreading-java.html)

Multithreading in Java Tutorial with Examples
(/multithreading-java.html)

JAVA TUTORIALS

(/prime-number-program-java.html)
(/prime-number-program-java.html)

Prime Number From 1 to 100 Program in Java
(/prime-number-program-java.html)

JAVA TUTORIALS

(/convert-char-string-java.html)
(/convert-char-string-java.html)

How to Convert Char to String & String to char in Java
(/convert-char-string-java.html)

JAVA TUTORIALS

(/string-charat-method-java.html)
(/string-charat-method-java.html)

String charAt() Method in Java with Example
(/string-charat-method-java.html)

JAVASCRIPT

(/typescript-vs-javascript.html)
(/typescript-vs-javascript.html)

Typescript vs JavaScript: What's the Difference?
(/typescript-vs-javascript.html)

Java tutorials

Java tutorials

22) toLowercase() & toUpperCase() (/java-string-tolowercase-methods.html)

23) Convert a Java String to Integer (/convert-string-to-integer.html)

24) HashMaps in Java (/working-with-hashmaps.html)

25) Command-line (/java-command-line-arguments.html)

26) This Keyword (/java-this-keyword.html)

¶

(https://www.facebook.com/guru99com/) (https://twitter.com/guru99com) (https://www.youtube.com/channel/UC19i1XD6k88KqHlET8atqFQ)

(https://forms.aweber.com/form/46/724807646.htm)

About

About US (/about-us.html)
Advertise with Us (/advertise-us.html)
Write For Us (/become-an-instructor.html)
Contact US (/contact-us.html)

Career Suggestion

SAP Career Suggestion Tool (/best-sap-module.html)
Software Testing as a Career (/software-testing-career-complete-guide.html)
Certificates (/certificate-it-professional.html)

Interesting

Books to Read! (/books.html)
Suggest a Tutorial
Blog (/blog/)
Quiz (/tests.html)
Review (/best-ergonomic-mouse.html)

Execute online