# Inheritance in Java

* **Inheritance in Java** is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of [OOPs](https://www.javatpoint.com/java-oops-concepts) (Object Oriented programming system).
* The idea behind inheritance in Java is that you can create new [classes](https://www.javatpoint.com/object-and-class-in-java) that are built upon existing classes.
* When you inherit from an existing class, you can reuse methods and fields of the parent class.
* Moreover, you can add new methods and fields in your current class also.
* Inheritance represents the **IS-A relationship** which is also known as a *parent-child* relationship.

### Why use inheritance in java

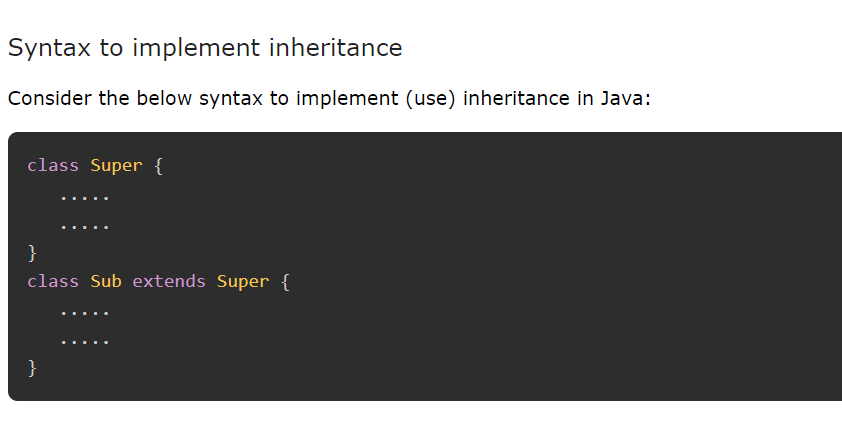
* For Method Overriding (so runtime polymorphism can be achieved).
* For Code Reusability.

### Terms used in Inheritance

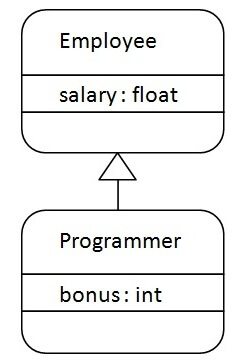
* **Class:** A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.
* **Sub Class/Child Class:** Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
* **Super Class/Parent Class:** Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.
* **Reusability:** As the name specifies, reusability is a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new class. You can use the same fields and methods already defined in the previous class.

## Implementation of Java Inheritance

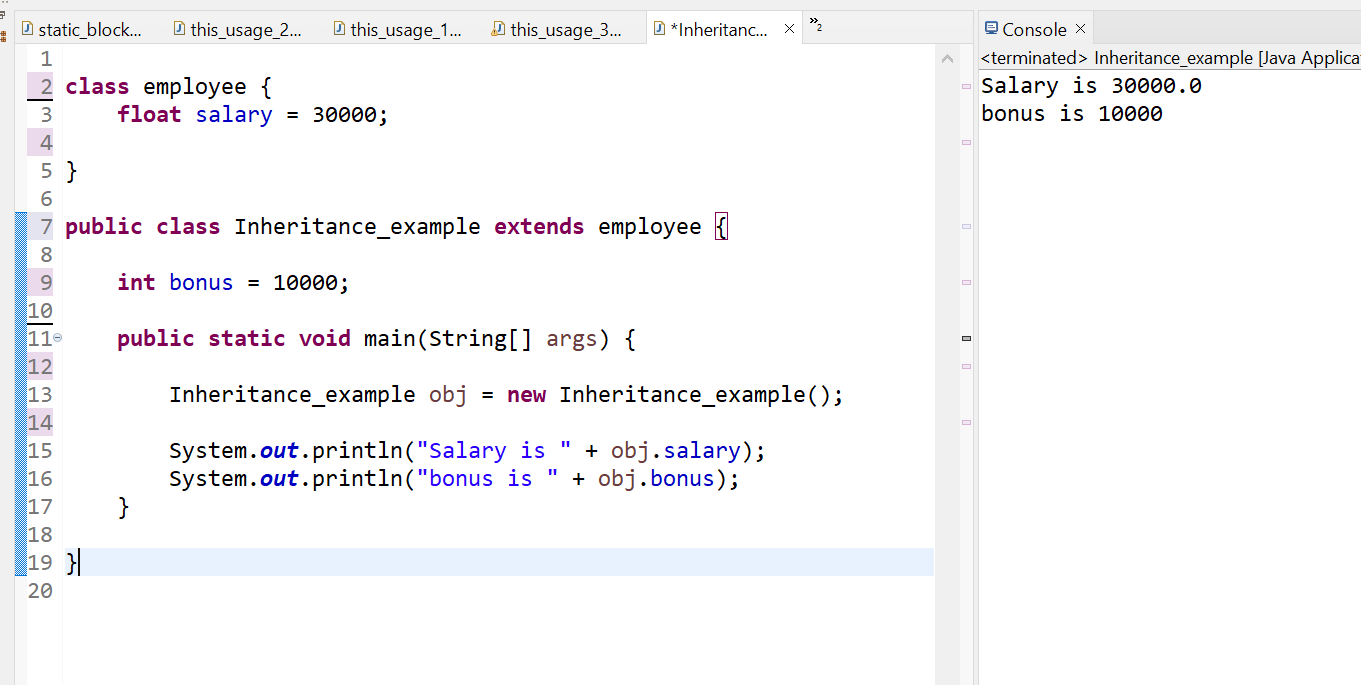
To implement (use) inheritance in Java, the extends keyword is used. It inherits the properties (attributes or/and methods) of the base class to the derived class. The word



* The **extends keyword** indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.
* In the terminology of Java, a class which is inherited is called a parent or superclass, and the new class is called child or subclass.
* Example

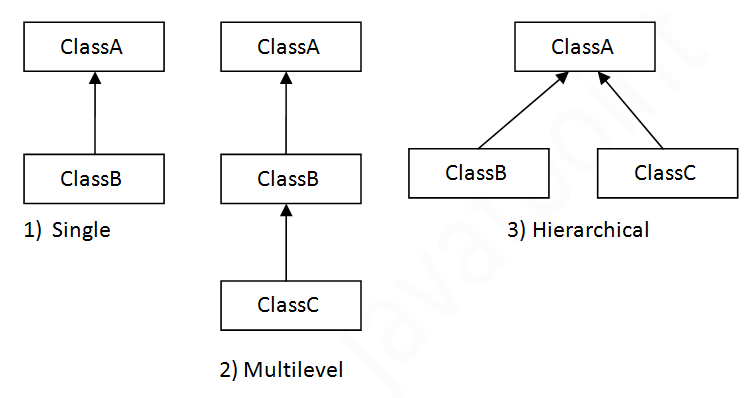


As displayed in the above figure, Programmer is the subclass and Employee is the superclass. The relationship between the two classes is **Programmer IS-A Employee**. It means that Programmer is a type of Employee.

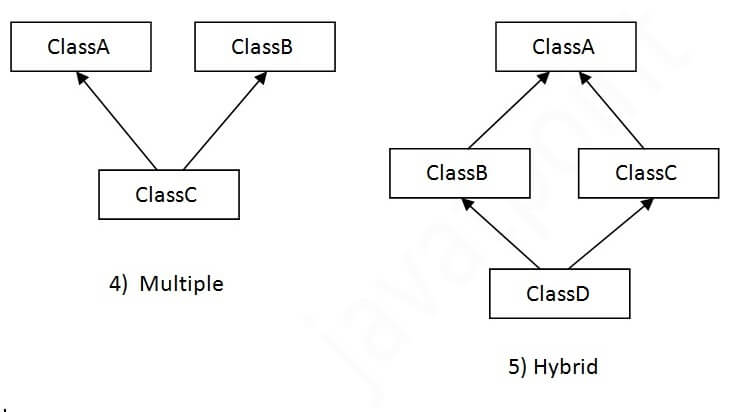


## Types of inheritance in java

On the basis of class, there can be three types of inheritance in java: single, multilevel and hierarchical.

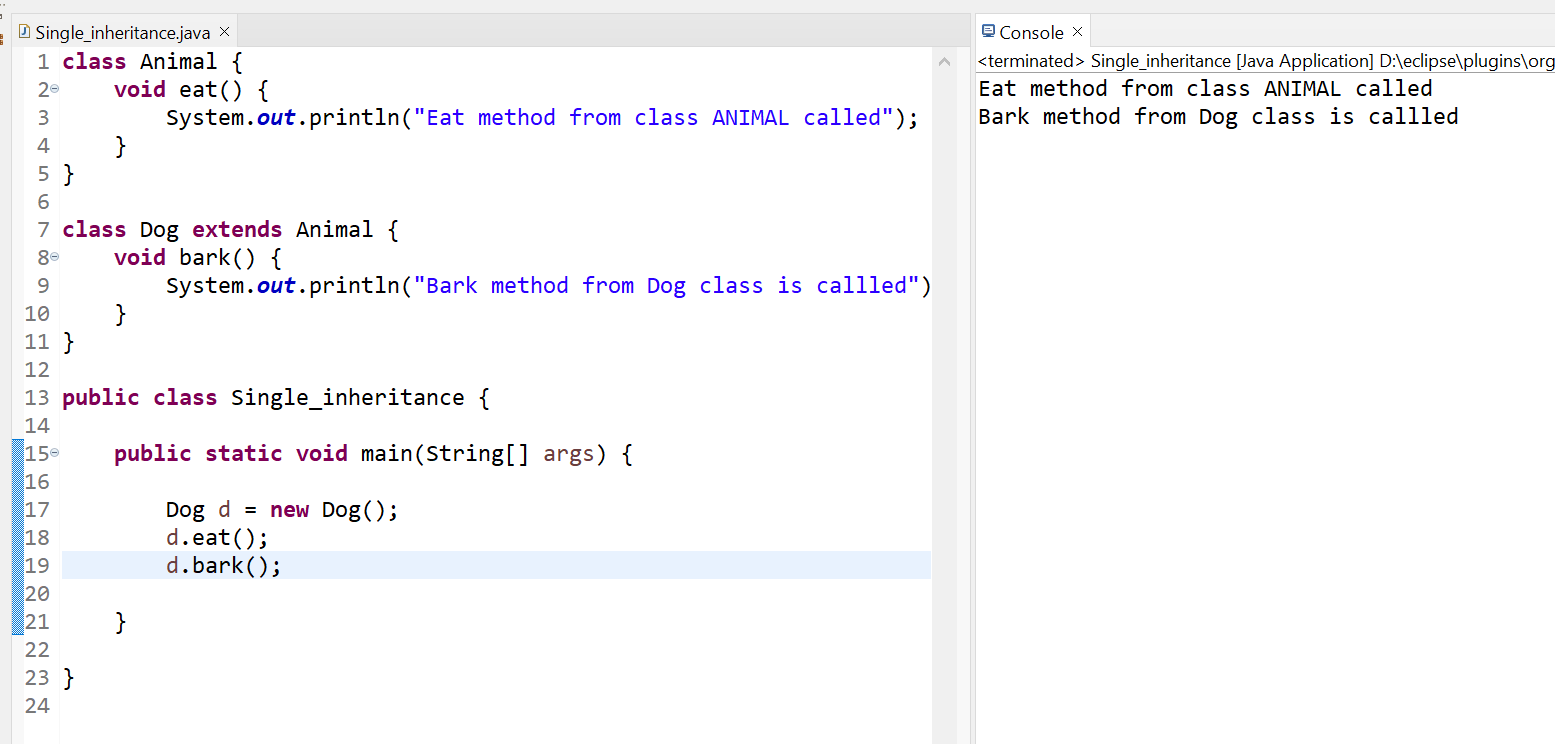


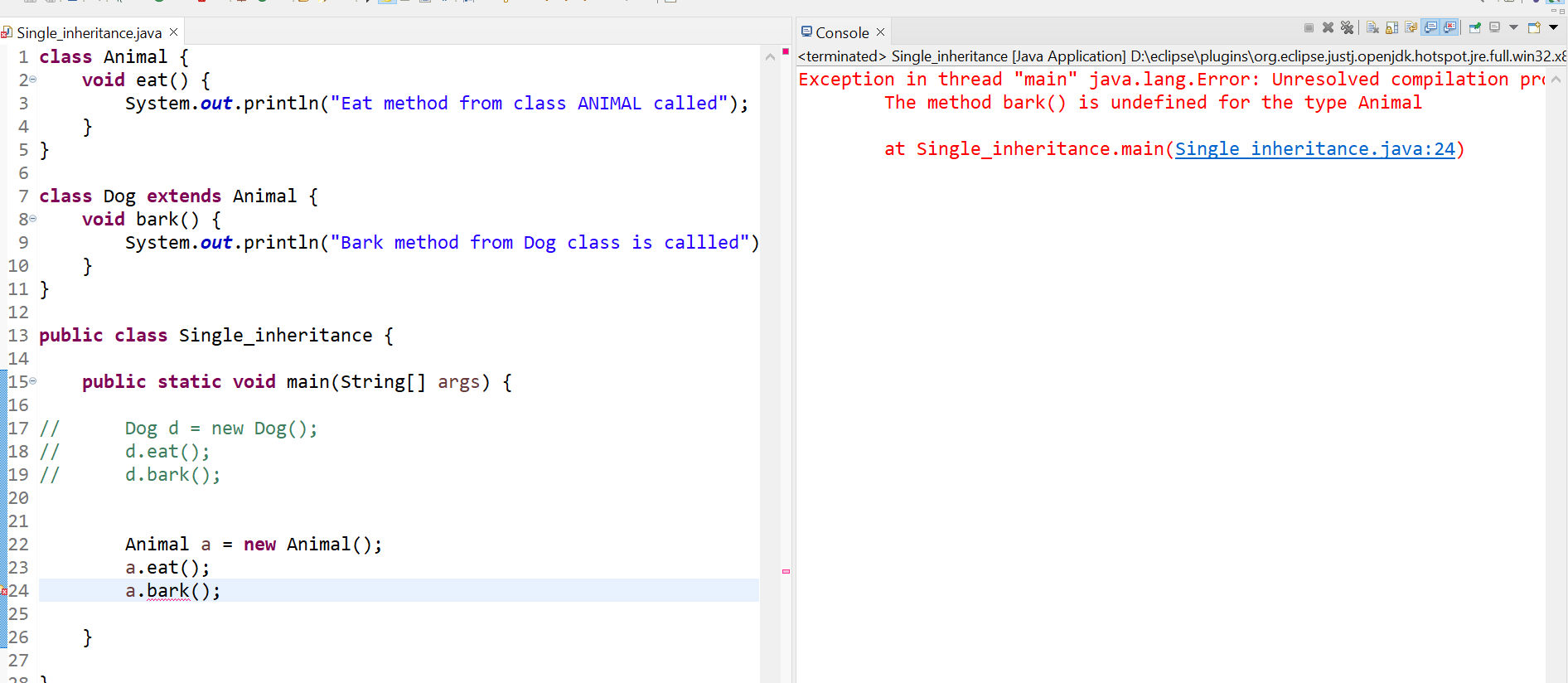
**Note ; Multiple inheritance is not supported in Java through class.When one class inherits multiple classes, it is known as multiple inheritance.**

****

## Single Inheritance Example

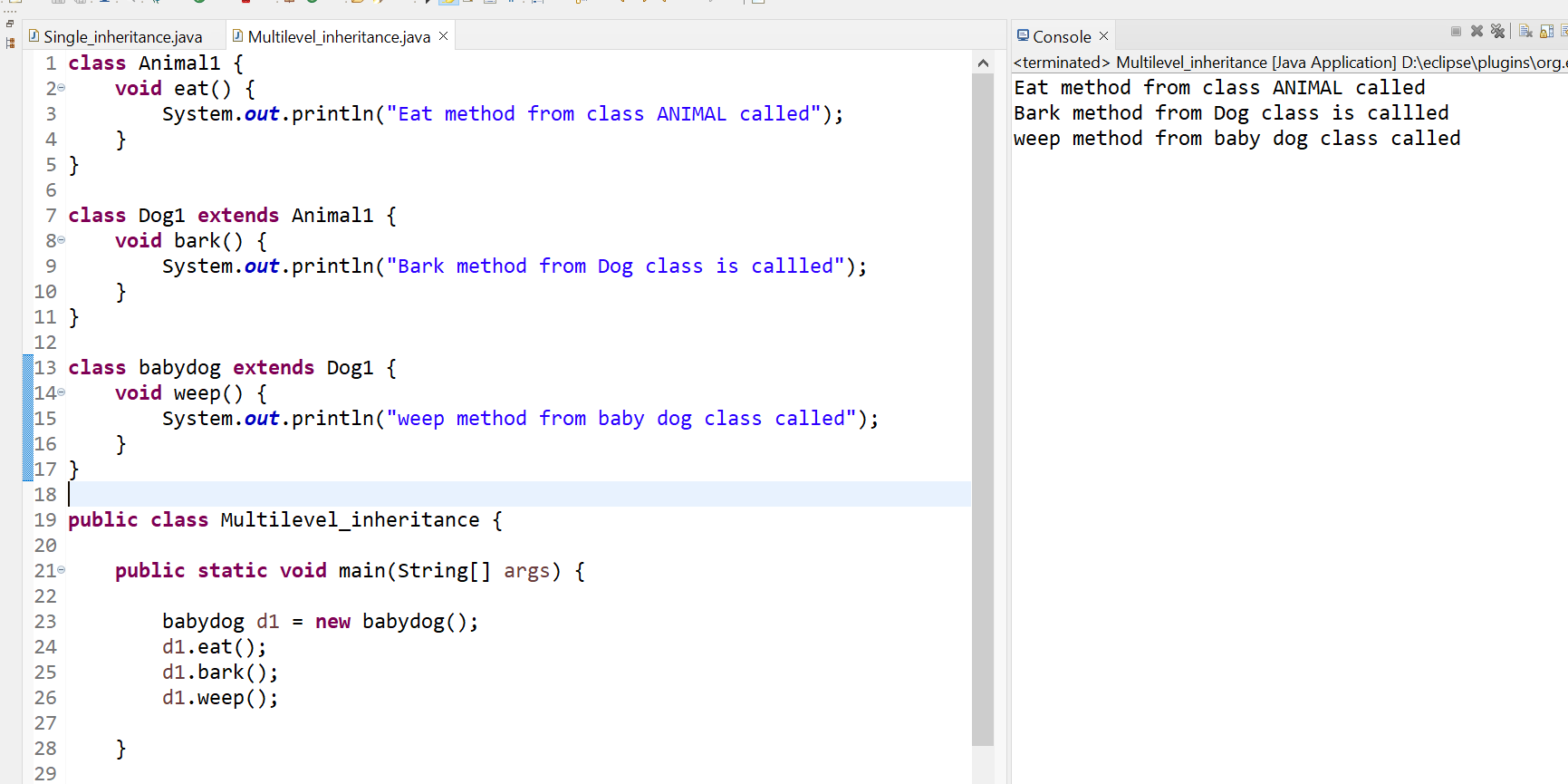
When a class inherits another class, it is known as a *single inheritance*. In the example given below, Dog class inherits the Animal class, so there is the single inheritance.

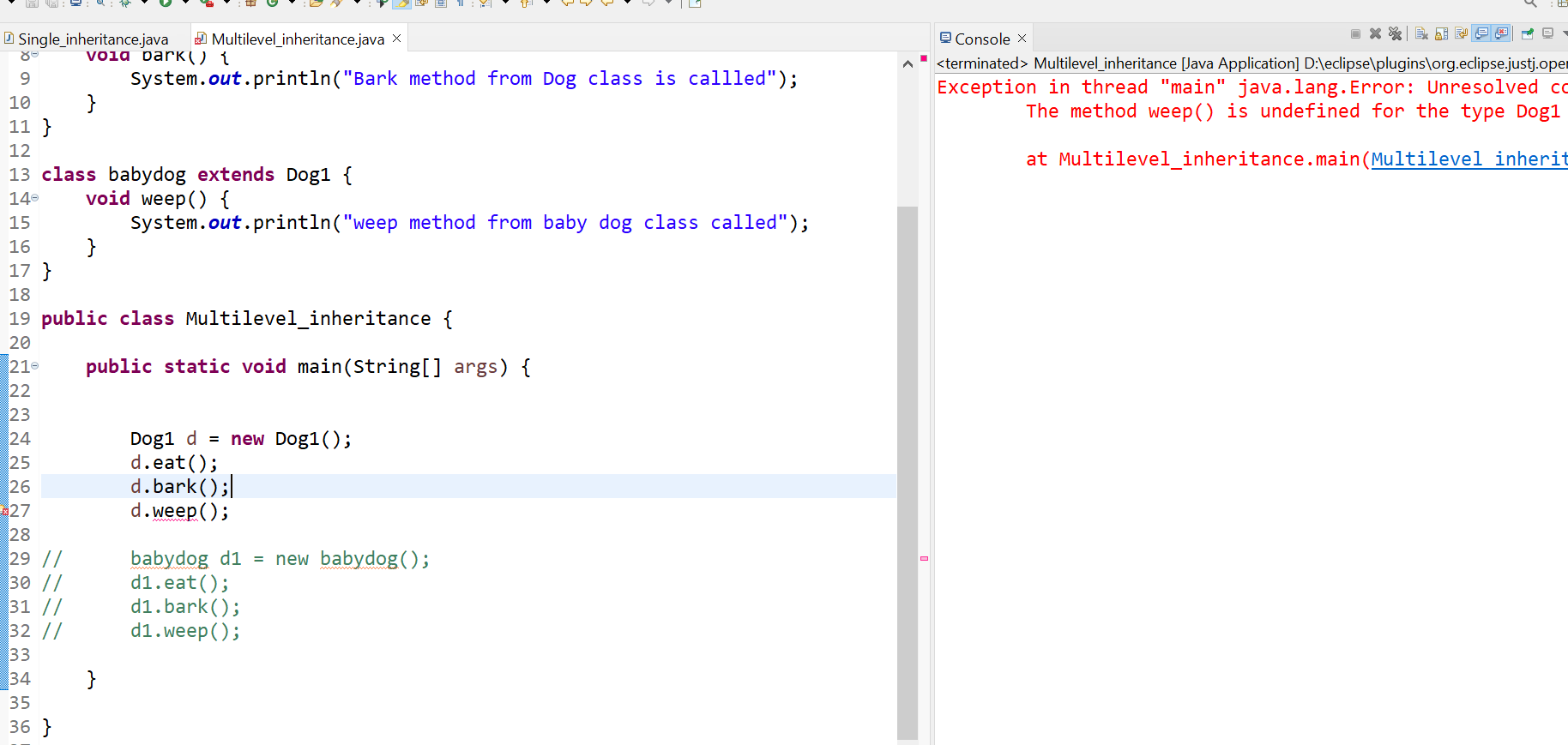




## Multilevel Inheritance Example

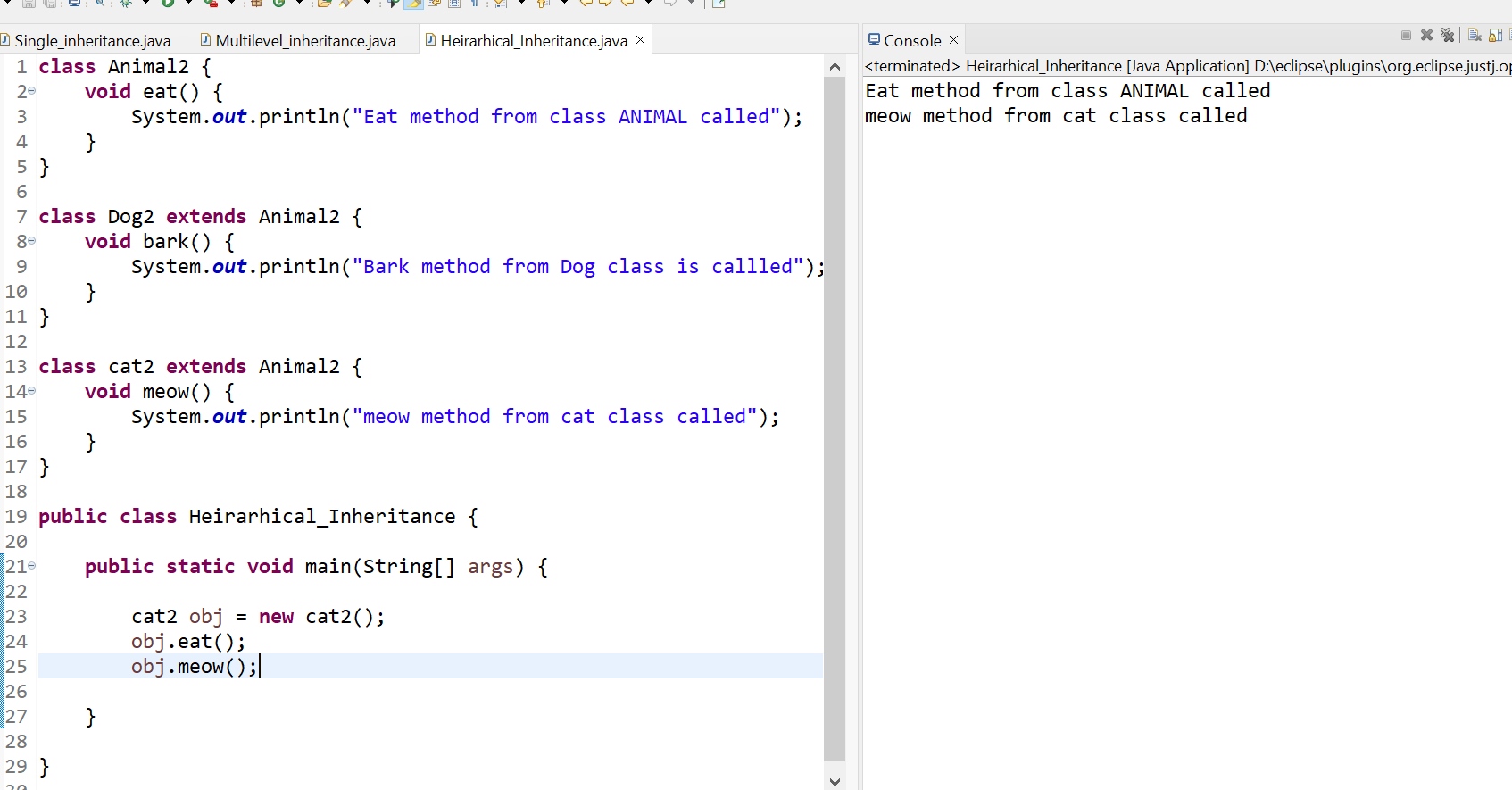
When there is a chain of inheritance, it is known as *multilevel inheritance*. As you can see in the example given below, BabyDog class inherits the Dog class which again inherits the Animal class, so there is a multilevel inheritance.

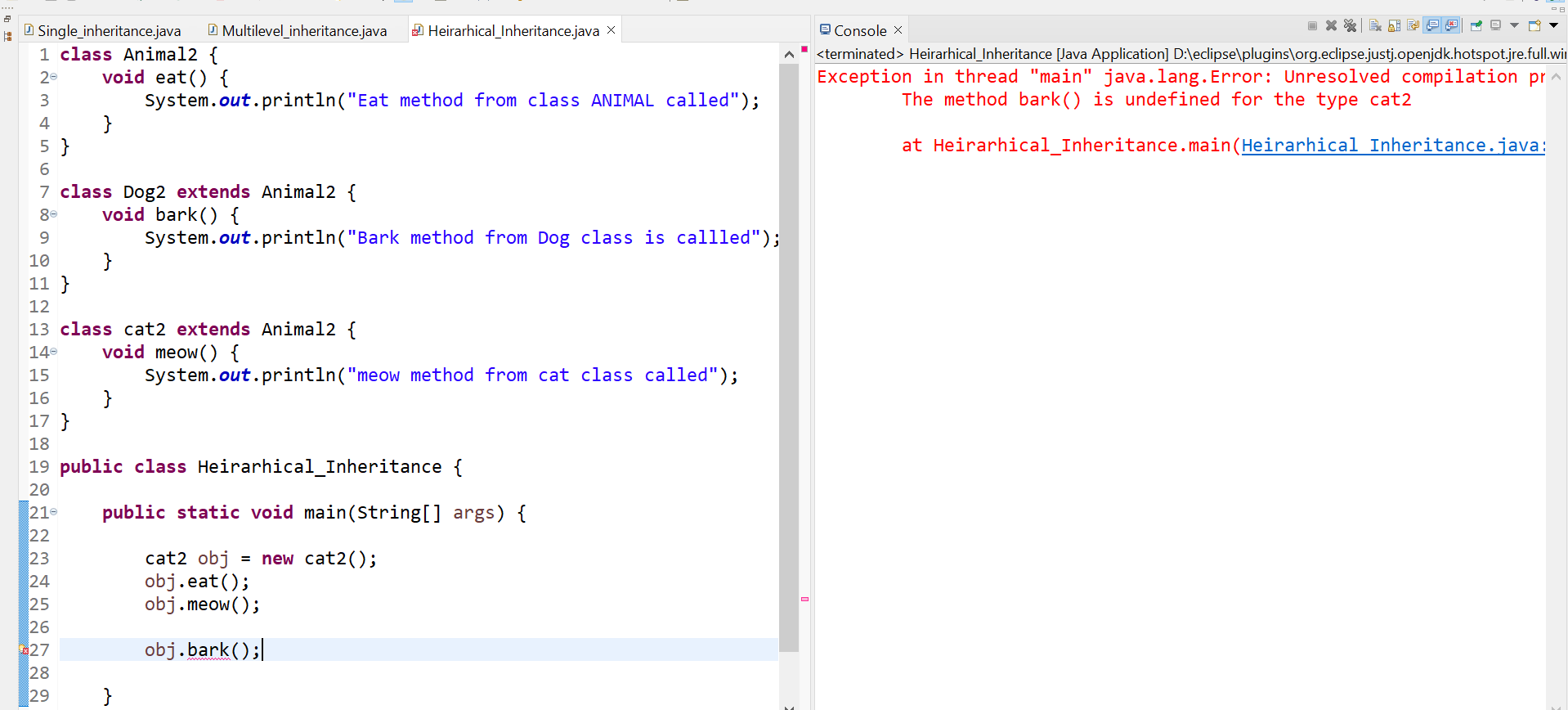




## Hierarchical Inheritance Example

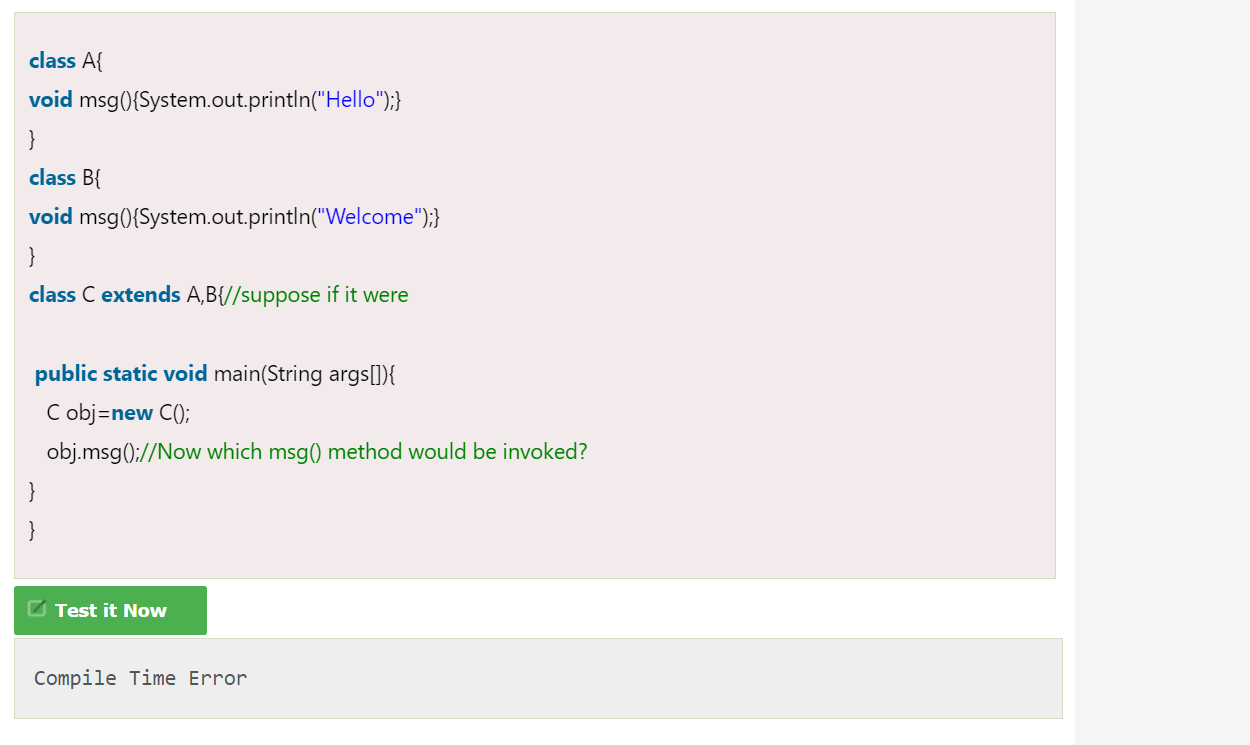
When two or more classes inherits a single class, it is known as *hierarchical inheritance*. In the example given below, Dog and Cat classes inherits the Animal class, so there is hierarchical inheritance.





## Q) Why multiple inheritance is not supported in java?

* To reduce the complexity and simplify the language, multiple inheritance is not supported in java.
* Consider a scenario where A, B, and C are three classes. The C class inherits A and B classes. If A and B classes have the same method and you call it from child class object, there will be ambiguity to call the method of A or B class.
* Since compile-time errors are better than runtime errors, Java renders compile-time error if you inherit 2 classes. So whether you have same method or different, there will be compile time error.



## Java Inheritance: The super Keyword

* The super keyword is similar to this keyword. Following are the scenarios where the super keyword is used.
* It is used to differentiate the members of superclass from the members of subclass, if they have same names.
* It is used to invoke the superclass constructor from subclass.

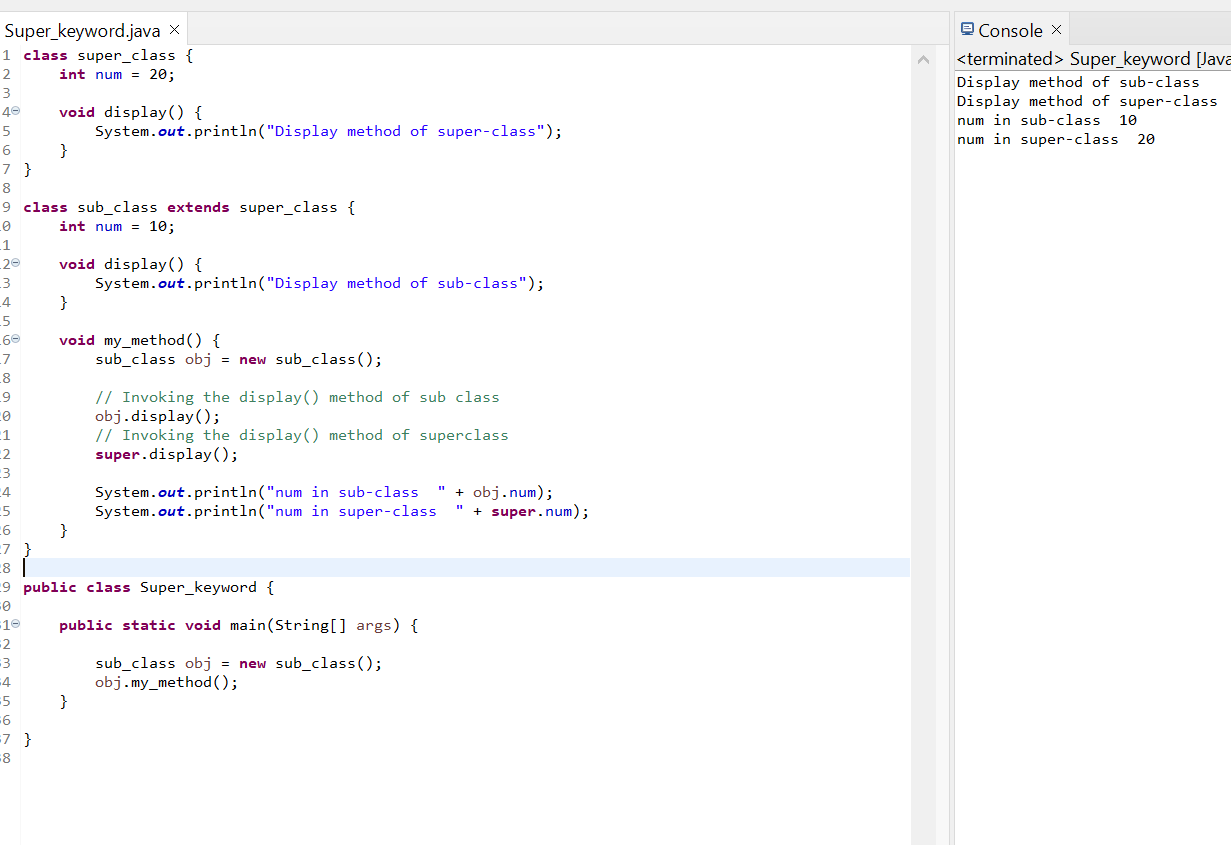
### Differentiating the Members

If a class is inheriting the properties of another class. And if the members of the superclass have the names same as the sub class, to differentiate these variables we use super keyword as shown below.



Example :

In the given program, you have two classes namely Sub\_class and Super\_class, both have a method named display() with different implementations, and a variable named num with different values. We are invoking display() method of both classes and printing the value of the variable num of both classes. Here you can observe that we have used super keyword to differentiate the members of superclass from subclass.



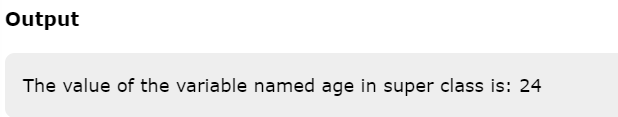
## Invoking Superclass Constructor

If a class is inheriting the properties of another class, the subclass automatically acquires the default constructor of the superclass. But if you want to call a parameterized constructor of the superclass, you need to use the super keyword as shown below.



The program given in this section demonstrates how to use the super keyword to invoke the parametrized constructor of the superclass. This program contains a superclass and a subclass, where the superclass contains a parameterized constructor which accepts a integer value, and we used the super keyword to invoke the parameterized constructor of the superclass.





# Aggregation in Java

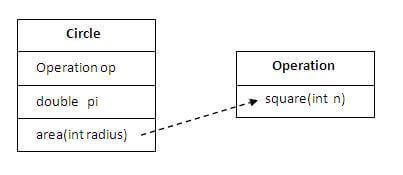
If a class have an entity reference, it is known as Aggregation. Aggregation represents HAS-A relationship.

Consider a situation, Employee object contains many informations such as id, name, emailId etc. It contains one more object named address, which contains its own informations such as city, state, country, zipcode etc. as given below.

1. **class** Employee{
2. **int** id;
3. String name;
4. Address address;//Address is a class
5. ...
6. }

In such case, Employee has an entity reference address, so relationship is Employee HAS-A address.

### Why use Aggregation?



| In this example, we have created the reference of Operation class in the Circle class. When use Aggregation?  * Code reuse is also best achieved by aggregation when there is no is-a relationship. * Inheritance should be used only if the relationship is-a is maintained throughout the lifetime of the objects involved; otherwise, aggregation is the best choice.  Understanding meaningful example of Aggregation In this example, Employee has an object of Address, address object contains its own informations such as city, state, country etc. In such case relationship is Employee HAS-A address.       **Java - Polymorphism** |
| --- |

For Videos Join Our Youtube Channel: [Join Now](https://bit.ly/2FOeX6S)

### Feedback

* Send your Feedback to feedback@javatpoint.com

## Help Others, Please Share

facebook twitter pinterest

## Learn Latest Tutorials

[Splunk](https://www.javatpoint.com/splunk)

[SPSS](https://www.javatpoint.com/spss)

[Swagger](https://www.javatpoint.com/swagger)

[Transact-SQL](https://www.javatpoint.com/t-sql)

[Tumblr](https://www.javatpoint.com/tumblr)

[ReactJS](https://www.javatpoint.com/reactjs-tutorial)

[Regex](https://www.javatpoint.com/regex)

[Reinforcement Learning](https://www.javatpoint.com/reinforcement-learning)

[R Programming](https://www.javatpoint.com/r-tutorial)

[RxJS](https://www.javatpoint.com/rxjs)

[React Native](https://www.javatpoint.com/react-native-tutorial)

[Python Design Patterns](https://www.javatpoint.com/python-design-pattern)

[Python Pillow](https://www.javatpoint.com/python-pillow)

[Python Turtle](https://www.javatpoint.com/python-turtle-programming)

[Keras](https://www.javatpoint.com/keras)

## Preparation

[Aptitude](https://www.javatpoint.com/aptitude/quantitative)

[Reasoning](https://www.javatpoint.com/reasoning)

[Verbal Ability](https://www.javatpoint.com/verbal-ability)

[Interview Questions](https://www.javatpoint.com/interview-questions-and-answers)

[Company Questions](https://www.javatpoint.com/company-interview-questions-and-recruitment-process)

## Trending Technologies

[Artificial Intelligence](https://www.javatpoint.com/artificial-intelligence-ai)

[AWS](https://www.javatpoint.com/aws-tutorial)

[Selenium](https://www.javatpoint.com/selenium-tutorial)

[Cloud Computing](https://www.javatpoint.com/cloud-computing)

[Hadoop](https://www.javatpoint.com/hadoop-tutorial)

[ReactJS](https://www.javatpoint.com/reactjs-tutorial)

[Data Science](https://www.javatpoint.com/data-science)

[Angular 7](https://www.javatpoint.com/angular-7-tutorial)

[Blockchain](https://www.javatpoint.com/blockchain-tutorial)

[Git](https://www.javatpoint.com/git)

[Machine Learning](https://www.javatpoint.com/machine-learning)

[DevOps](https://www.javatpoint.com/devops)

## B.Tech / MCA

[DBMS](https://www.javatpoint.com/dbms-tutorial)

[Data Structures](https://www.javatpoint.com/data-structure-tutorial)

[DAA](https://www.javatpoint.com/daa-tutorial)

[Operating System](https://www.javatpoint.com/operating-system)

[Computer Network](https://www.javatpoint.com/computer-network-tutorial)

[Compiler Design](https://www.javatpoint.com/compiler-tutorial)

[Computer Organization](https://www.javatpoint.com/computer-organization-and-architecture-tutorial)

[Discrete Mathematics](https://www.javatpoint.com/discrete-mathematics-tutorial)

[Ethical Hacking](https://www.javatpoint.com/ethical-hacking)

[Computer Graphics](https://www.javatpoint.com/computer-graphics-tutorial)

[Software Engineering](https://www.javatpoint.com/software-engineering)

[Web Technology](https://www.javatpoint.com/html-tutorial)

[Cyber Security](https://www.javatpoint.com/cyber-security-tutorial)

[Automata](https://www.javatpoint.com/automata-tutorial)

[C Programming](https://www.javatpoint.com/c-programming-language-tutorial)

[C++](https://www.javatpoint.com/cpp-tutorial)

[Java](https://www.javatpoint.com/java-tutorial)

[.Net](https://www.javatpoint.com/net-framework)

[Python](https://www.javatpoint.com/python-tutorial)

[Programs](https://www.javatpoint.com/programs-list)

[Control System](https://www.javatpoint.com/control-system-tutorial)

[Data Mining](https://www.javatpoint.com/data-mining)

[Data Warehouse](https://www.javatpoint.com/data-warehouse)