**Q: What is Inner Class in Java?  
A:**In Java similar methods and variables of a class, we can have a class as member of another class . Declaring a class within another is allowed in Java. The class written within is called the nested class, and the class that holds the inner class is called the outer class. We use inner classes to logically group classes and interfaces in one place so that it can be more readable and maintainable.

class Java\_Outer\_Demo {

class Java\_Inner\_Demo {

}

}

**Q: What are the advantages of using Inner Class in Java?  
A:**The Advantages of using Polymer are as follows-

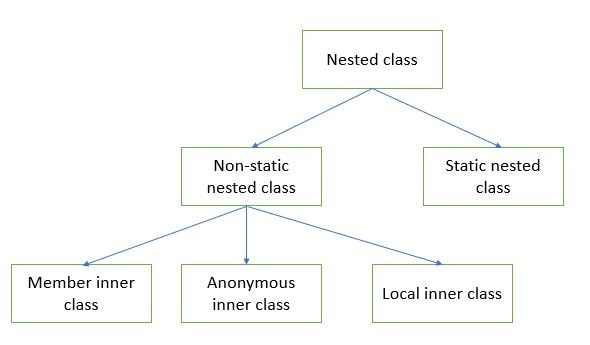
* Nested classes represent a special type of relationship that is it can access all the members (data members and methods) of outer class including private.
* Nested classes are used to develop more readable and maintainable code because it logically group classes and interfaces in one place only.
* Code Optimization: It requires less code to write.

**Q: What are the types of nested classes?  
A:**Nested classes can be divided into two types -

* Non-static nested classes - These are the non-static members of a class.
* Static nested classes - These are the static members of a class.

Non-static class can be further classified into 3 types

* Member inner class
* Anonymous inner class
* Local inner class

  
**Q: What is anonymous inner class in Java?  
A:**It is an inner class without a name and for which only a single object is created. An anonymous inner class can be useful when making an instance of an object with certain "extras"" such as overloading methods of a class or interface, without having to actually subclass a class.  
Anonymous inner classes are useful in writing implementation classes for listener interfaces in graphics programming. Anonymous inner class can be created in 2 ways-

* Class (may be abstract or concrete)
* Interface

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

JFileChooser fileChooser = new JFileChooser();

int returnVal = fileChooser.showOpenDialog(null);

if (returnVal == JFileChooser.APPROVE\_OPTION) {

System.out.println(fileChooser.getSelectedFile().getName());

}

}

});

**Q: Why can outer Java classes access inner class private members?  
A:** The inner class is just a way to cleanly separate some functionality that really belongs to the original outer class. They are intended to be used when you have 2 requirements: Some piece of functionality in your outer class would be most clear if it was implemented in a separate class. Even though it's in a separate class, the functionality is very closely tied to way that the outer class works. Given these requirements, inner classes have full access to their outer class. Since they're basically a member of the outer class, it makes sense that they have access to methods and attributes of the outer class -- including privates.