* What object knows is the variables, what object does is the methods.
* An object can only use instance variable, you can’t call a local variable using an object.
* An object don’t have direct access to local variables, if you want to use local variables, you should call the method in which a local variable is a part of it , and the method will use the local variable.
* If you want to set the value when the object is created you can use constructor . And if you want to set the value later you can use setter

System class has static object of PrintStream class which is declared in System class as out and the println() is the method of PrintStream class. So we can access static object as System. out and the println() is the method of PrintStream class.

( Use Ctrl+B to show the predefined code in intellij idea ) .

Eg: Println\_PrintStream

* You cannot access non static method inside a static method ( main method ) , so you need to create a object of class and with that object the non static method can be called in the static method

Inner Classes:

* Inner class (or) nested class is a class that is declared inside another class (or) interface.
* Inner class can access all the members of outer class, including private data members and methods as well.
* We can make inner class as static, but outer class cannot be static
* We cannot have static method inside a nested inner class.

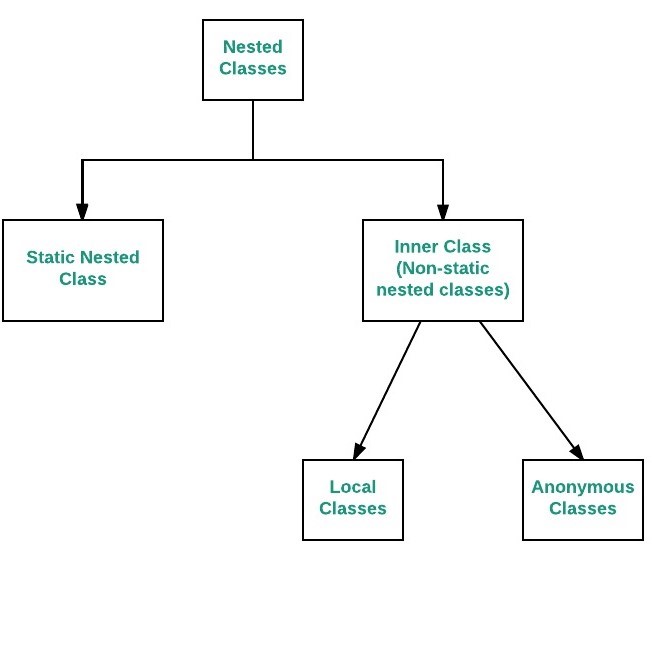
Need of Inner class:

Sometimes users need to program a class in such a way so that no other class can access it. Therefore, it would be better if you include it within other classes.

Difference between Nested and Inner classes:

static class declared inside another class is called static Nested class.

non - static class declared inside another class is called inner class.



Java Member Inner class:

A non-static class that is declared inside a class but outside a method is called java member inner class. It is also known as regular inner class. it can be declared with access modifiers like public, private, protected and default.

It can access private instance variable of outer class.

Eg: Java\_Member\_Inner\_Class

// go through the code

Anonymous Inner Class:

In java it is possible to create a nested class without giving any name. A nested class that doesn’t have any name is called Anonymous class.

An Anonymous class must be declared inside another class hence it is known as anonymous inner class.

Anonymous classes usually extend subclasses (or) implement interfaces

Anonymous class enables you to declare and instantiate class at the same time, they are like local classes except that they do not have any name.

It should be used to override method of a class (or) interface

Anonymous inner class can be created in 2 ways

1. Class (abstract (or) concrete)
2. Interface

Internally anonymous class extends the outer class

Internally anonymous class implements the interface

.class is also produced for anonymous collage, and jvm gives a name to anonymous class

Eg: Anonymous\_Inner\_Class

Eg: Anonymous\_Inner\_Class\_Using\_Interface

Java Local Inner Class:

Class created inside a method is called local inner class. They are defined inside a block, Generally they are defined inside a method. Sometimes this block can be of for loop (or) an if clause.

They belong to the block they are defined within, due to which local inner classes does not have access specifiers. They can be marked as final (or) abstract.

Local inner class can’t use local variable of outer method until the local variable is declared as final (until jdk 1.8 (or) java 8)

Since 1.8 it is possible without using final

Eg: Local\_Inner\_Class\_Instance\_Variable

Eg: Local\_Inner\_Class\_Local\_Variable

Static Class:

In java only nested class is allowed to be static

Like regular classes static nested classes can have both static and non- static fields and methods in it.

static class is declared inside a class, it cannot access non static data members and methods . It can be accessed by outer class name.

It can only access static data members and methods of outer class including private.

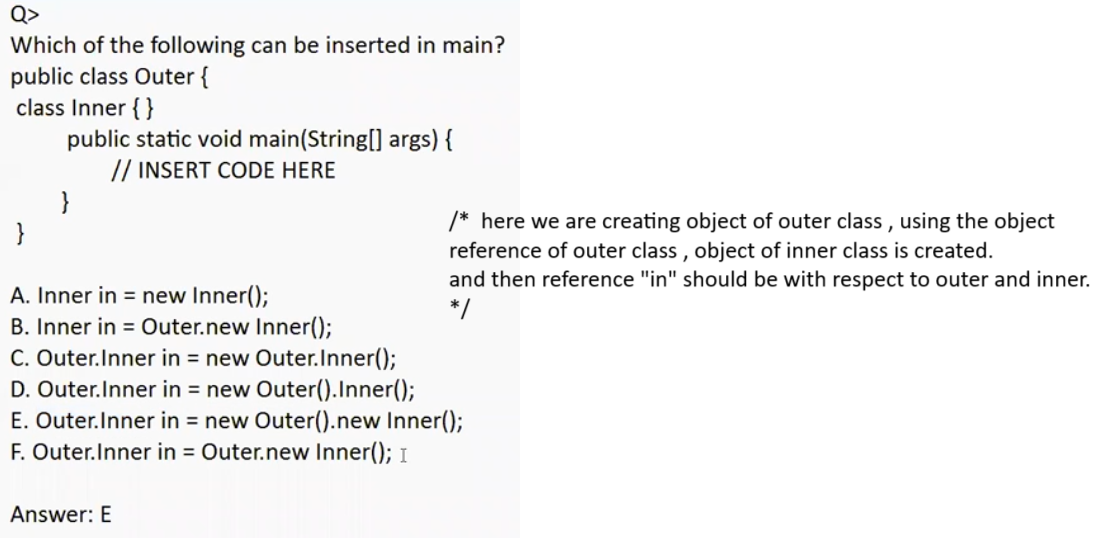
Eg: Static\_Nested\_Class\_With\_Instance\_Method

Eg: Static\_Nested\_Class\_With\_Static\_Method

// once interface concept is completed go through nested interface

Eg: Innerclass\_Eg1

// go through the code



Eg: Inner\_Class\_Eg2

// go through the code

Eg: Inner\_Class\_Eg3

// go through the code