IQ :

Why we should declare a constructor with private Access modifier ?

We can declare a Constructor with private access modifier so the accessibility level of the constructor will be from the same class only that means we can create the object from the BLC class only, Object creation is not possible from ELC class.

1) If static method and static variables are available
2) Once we declare a constructor with private AM then only same class can create the Object, This concept is known as Singleton class

---

Instance Block OR Instance Initializer OR Non static block :

Instance Block OR Instance Initializer OR Non static block :

An instance block is a very special block in java which is automatically executed at the time of creating the Object.

Example :

```
{
  System.out.println("It is a Non static block");
}
```

Every time we will create an Object then instance block will be executed.

NSBDemo1.java

```
package com.ravi.non_static_block;

class Test
{
    {
        System.out.println("NSB");
    }
}

public class NSBDemo1
{
    public static void main(String[] args)
    {
        new Test();
        new Test();

    }

}
```

A non static block is executed before the Constructor body execution.

The first line of any constructor is reserved for this() OR super(). If a contsructor contains super() as a first line of constructor then that constructor 2nd line is reserved for Non static block.

Case 1 :

Test.java
```
public class Test
{
   public Test()
   {
     super();
     System.out.println("Constructor");
   }

   {
     System.out.println("NSB");
   }
}
```

javac

Test.class
```
public class Test
{
   public Test()
   {
      super();  //1st line added by javac
      {
        System.out.println("NSB");  //2nd
      }
      System.out.println("Constructor");
   }
}
```