NON STATIC

NON STATIC :

- Any member declared in a class and not prefixed with a static modifier is known as a non-static member of a class.
- Non-static members belong to an instance of a class. Hence it is also known as an instance member or object member.
- The memory for the non-static variable is allocated inside the heap area(instance of a class).
- We can create any number of instances for a class.
- Non-static members will be allocated in every instance of a class.

NON STATIC MEMBERS :

- Non-static variable
- Non-static method
- Non-static initializers
- Constructors

NON STATIC VARIABLE :

A variable declared inside a class block and not prefixed with a static modifier is known as a non-static variable.

CHARACTERISTICS :

- We can’t use the non-static variable without creating an object.
- We can only use the non-static variable with the help of object reference.
- Non-static variables are assigned with default values during the object loading process.
- Multiple copies of non-static variables will be created (once for every object).

NON STATIC METHOD :

A method declared in a class block and not prefixed with a static modifier is known as a non-static method.

CHARACTERISTICS :

- A method block will get stored inside the method area and a reference of the method is stored inside the instance of a class [object].
- We can’t call the non-static method of a class without creating an instance of a class[object].
- We can’t access the non-static method directly with the help of class names.
- The non-static method can’t be accessed directly with their names inside the static context.

NON STATIC CONTEXT :

- The block which belongs to the non-static method and multi-line non-static initializer is known as non-static context.
- Inside a non-static context, we can use static and non-static members of the same class directly by using its name.

NON STATIC INITIALIZERS :

- Non-static initializers will execute during the loading process of an object.
- Non-static initializers will execute once for every instance of a class created. [object created].

PURPOSE OF NON STATIC INITIALIZERS :

Non-static initializers are used to execute the startup instructions for an object.

TYPES OF NON STATIC INITIALIZERS :

1. Single line non-static initializer
2. Multi-line non-static initializer

1. SINGLE LINE NON STATIC INITIALIZER :

Syntax to create single line non static initializers :

```
datatype variable = value / reference
```

2. MULTI LINE NON STATIC INITIALIZER :

Syntax to create multi line non static initializers :

```
{
// statements ;
}
```

NOTE :All the Non-static initializers will execute from top to bottom order for every object creation

Static Method	***Non static method***
It is declared in the class prefixed with static keyword	It is declared in class block not prefixed with static keyword.
Can be invoked with the help of classname or Object reference	Can be invoked only with the help of object reference
Can be invoked without creating object	Cannot be invoked without creating object
The block which belongs to static method is known as static context	The block which belongs to non-static method is known as non-static context
Inside static context we can use only static members of same class directly without any reference	Inside non-static context we can use both static and non-static members of same class directly without any reference

Static variable	***Non static variable***
It is declared in the class block prefixed with static keyword.	It is declared in the class block not prefixed with static keyword
Can be used with the help of class name as well as object reference	Can be used only with the help of object reference.
Can be used without creating object	Cannot be used without creating object
Static variable memory allocated in the class static area.	Non-static variable memory allocated in the object created in heap area.
Static variable will have only one copy shared by all the objects of class	Will have one copy allocated inside every object of the class. Hence it is shared memory.

Static initializer	***Non-static Initializer***
It is declared in the class block prefixed with static	It is declared in the class block not prefixed with static
Always gets executed when the class is loaded into the JRE.	Gets executed every time when an object is created.
Static block is considered as static context	Non-static block is considered as Non-static context.

this :

- It is a keyword.
- It is a non-static variable it holds the reference of a current executing object.

USES OF this :

- Used to access the members of the current object.
- It is used to give the reference of the current object.
- Reference of a current object can be passed from the method using the ‘this’ keyword.
- Calling a constructor of the same class is achieved with the help of this call statement.