

# CORE JAVA

Day-4

# PASS BY REFERENCE OF AN ARRAY

- In Java, arrays are treated as objects. When you create an array, you are actually creating an object in memory and the array variable holds the reference (or address) to this object.
- When you pass the array to a method, you are passing the reference to this object, not a new copy of the array.
- Therefore, if you modify the array inside the method, the changes are reflected in the original array since both the original and the parameter inside the method point to the same object in memory.



# STATIC BLOCKS

- In Java, a static block, also known as a static initialization block, is a block of code that gets executed only once when the class is loaded into memory.
- These blocks are typically used for static member initialization or executing a specific block of code when the class is first loaded, before any instances of the class are created or any static methods are called.
- Characteristics of Static Blocks
  - Static blocks are executed at class loading time. Before any object instantiation or any static method invocation, Java runs the static blocks.
  - They run only once. No matter how many objects of the class are created, static blocks execute only once.
  - Execution order is from top to bottom as they appear in the class. If a class contains multiple static blocks, they are executed in the order in which they appear in the class.

# STRING

- The String class in Java is used to create and manipulate strings.
- Strings created using the String class are immutable, meaning once a String object is created, its value cannot be changed.
- Any operation that seems to modify a String actually creates a new String object.
- This characteristic makes String less efficient for operations that require changing the content of the string repeatedly,

```
String s = "Hello";
```

```
s = s + " World"; // A new String object is created
```



# STRINGBUFFER

- StringBuffer is a peer class of String that provides much of the functionality of strings but is mutable.
- StringBuffer was designed to be used when there is a necessity to make a lot of modifications to strings of characters.
- Unlike String, StringBuffer can be modified over and over again without leaving behind a great litter of discarded string objects.
- The operations on StringBuffer are thread-safe, meaning they are synchronized and therefore safe to use in multi-threaded environments.

```
StringBuffer sb = new StringBuffer("Hello");  
sb.append(" World"); // Modifies the existing StringBuffer object
```

# STRINGBUILDER

- `StringBuilder` is similar to `StringBuffer` but with no guarantee of synchronization.
- This means that it is not thread-safe but in single-threaded scenarios, it is more efficient than `StringBuffer` due to the lack of synchronization overhead.
- If you're operating in a scenario where thread safety is not a concern, `StringBuilder` is generally the better choice for performance reasons.

```
StringBuilder sb = new StringBuilder("Hello");  
sb.append(" World"); // Modifies the existing StringBuilder object
```



# FINALIZE()

- The finalize() method in Java has been a part of Java since its earliest versions. It is defined in the java.lang.Object class, which means that it is available for use by any Java class.
- In Java 8, finalize() continues to be supported, but it has been deprecated starting from Java 9 .
- The finalize() method is called by the garbage collector on an object when garbage collection determines that there are no more references to the object.
- It is intended to give an object one last chance to clean up resources that might not yet have been released before the object is garbage collected.

# SYSTEM.GC()

- In Java, requesting the Java Virtual Machine (JVM) to perform garbage collection can be done using the `System.gc()` method.
- It's important to understand that calling this method does not guarantee that the garbage collection will actually occur; it merely suggests to the JVM that this is a good time to perform garbage collection.
- The JVM is free to ignore this request based on its own garbage collection algorithms and needs.