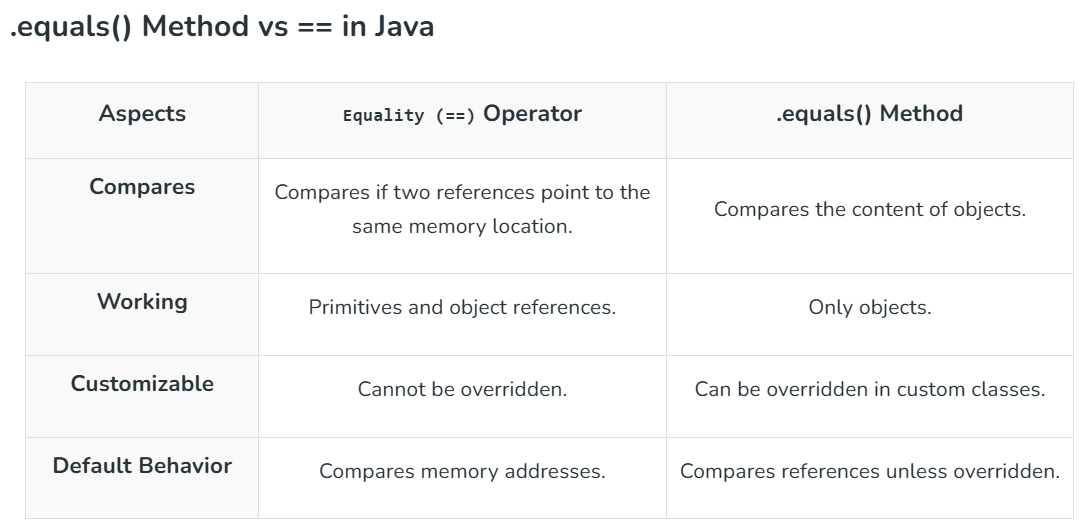
**JAVA QUESTIONS**

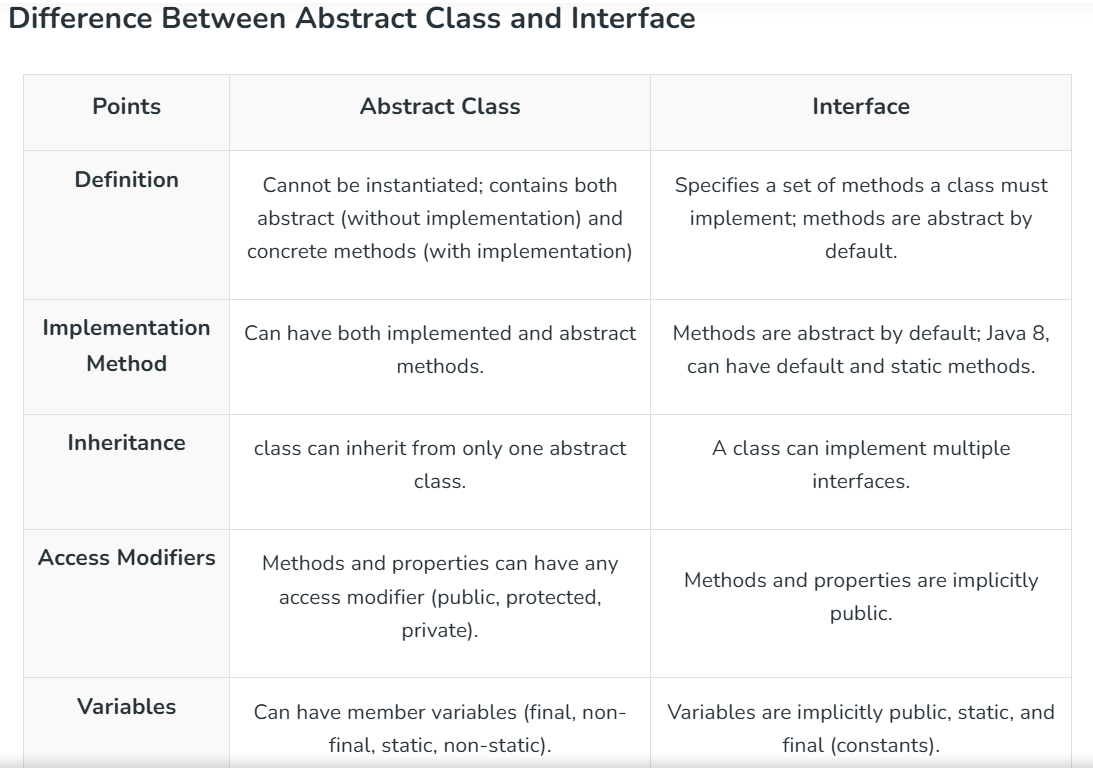
1. **Java String Pool and Memory**

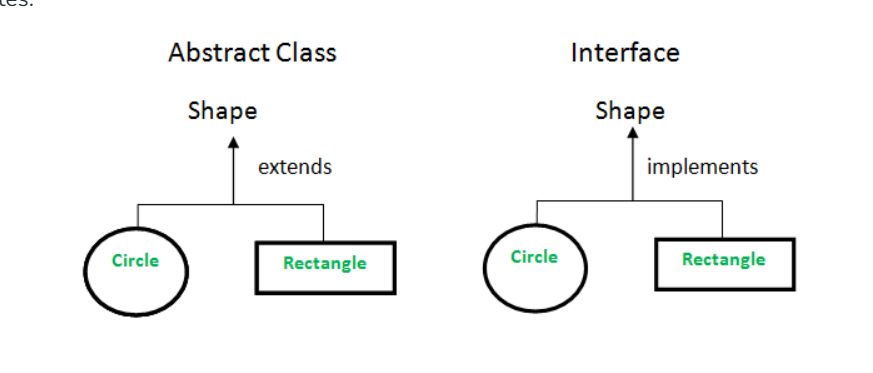
* **String Pool**: When two strings have the same content and are created without the new keyword, they point to the same memory location in the pool.
* [Heap](https://www.geeksforgeeks.org/what-is-a-memory-heap/): Strings created using the new keyword always allocate new memory.

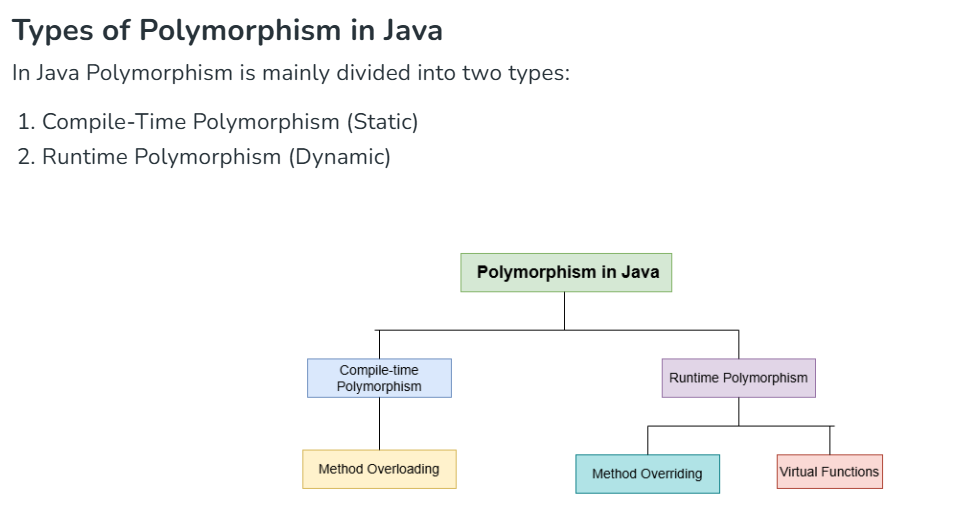
*String s1 = "HELLO"; // String pool*

*String s2 = new String("HELLO"); // Heap*



1. 



1. 

### Compile-Time Polymorphism

[Compile-Time Polymorphism](https://www.geeksforgeeks.org/compile-time-polymorphism-in-java/) in Java is also known as static polymorphism and also known as method overloading. This happens when multiple methods in the same class have the same name but different parameters.

**. Runtime Polymorphism**

[Runtime Polymorphism](https://www.geeksforgeeks.org/dynamic-method-dispatch-runtime-polymorphism-java/) in Java known as Dynamic Method Dispatch. It is a process in which a function call to the overridden method is resolved at Runtime. This type of polymorphism is achieved by Method Overriding. [Method overriding](https://www.geeksforgeeks.org/overriding-in-java/), on the other hand, occurs when a derived class has a definition for one of the member functions of the base class. That base function is said to be overridden.

**Method Overriding**

[Method overriding](https://www.geeksforgeeks.org/overriding-in-java/)in Java means when a subclass provides a specific implementation of a method that is already defined in its superclass. The method in the subclass must have the same name, return type, and parameters as the method in the superclass. Method overriding allows a subclass to modify or extend the behavior of an existing method in the parent class. This enables dynamic method dispatch, where the method that gets executed is determined at runtime based on the object's actual type.