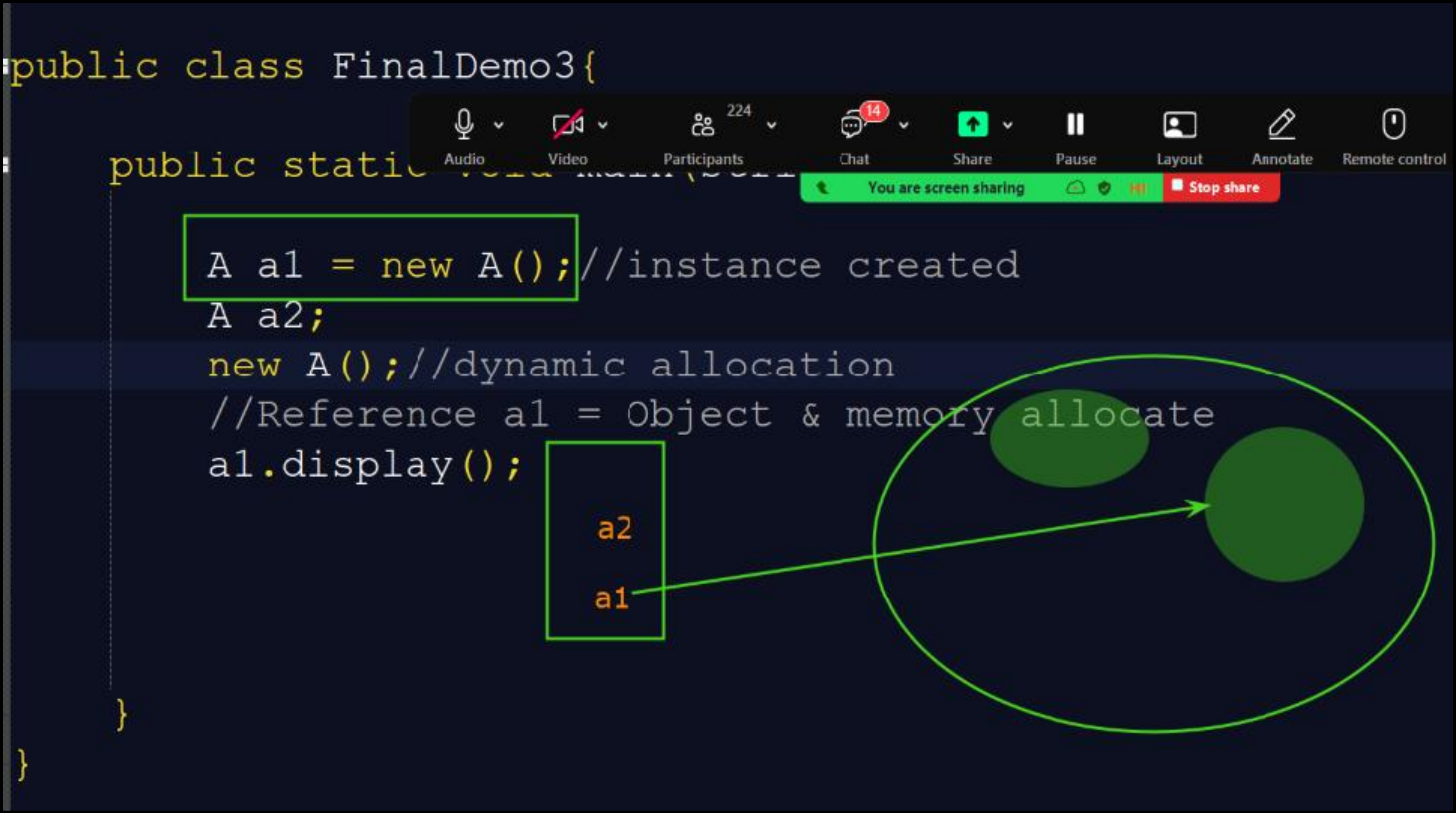
Object Oriented Programming with Java (OOPJ)

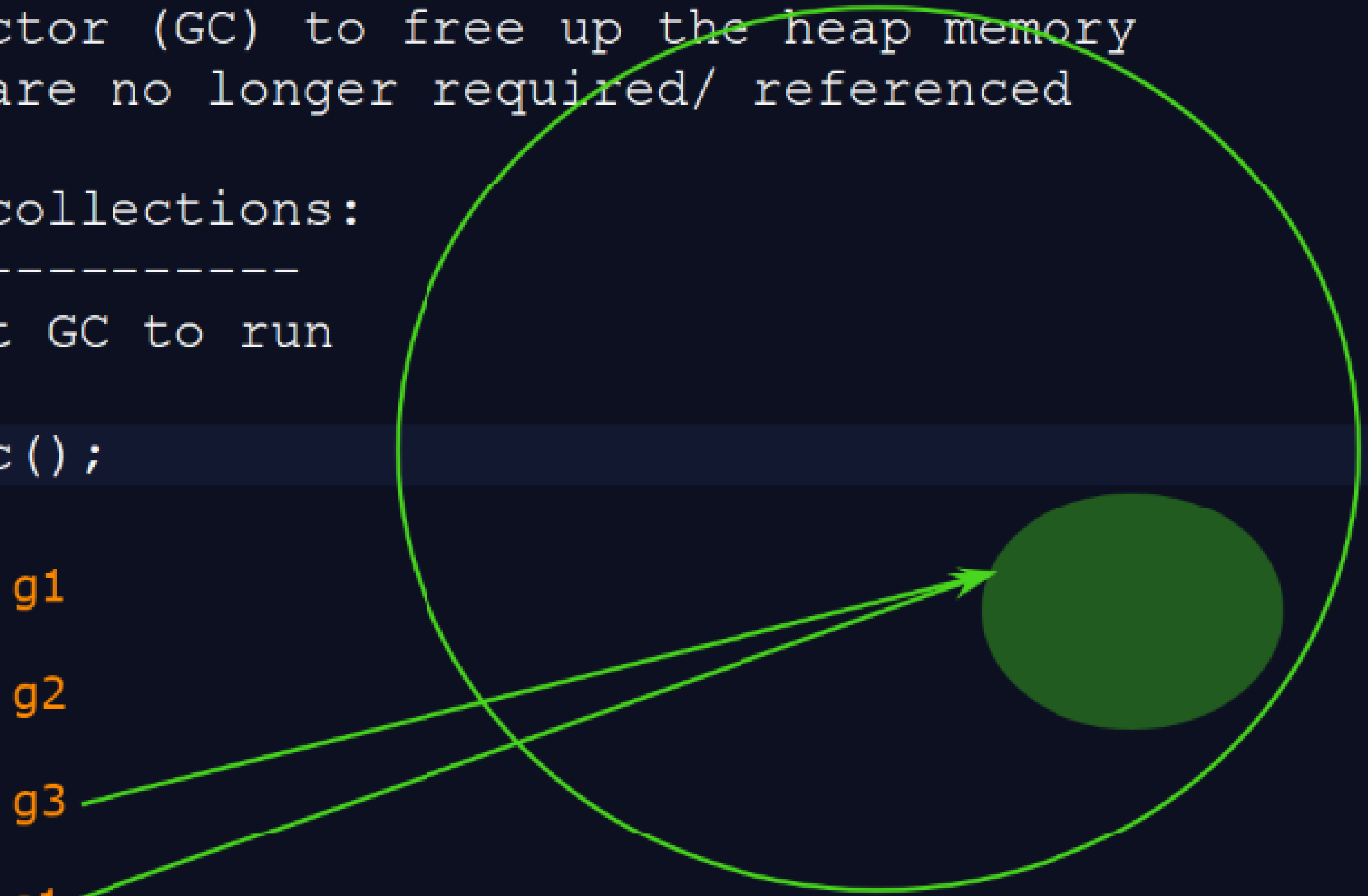
Session 10: Strings

Kiran Waghmare



```
public class FinalDemo3{

    public static

        A a1 = new A();//instance created
        A a2;
        new A();//dynamic allocation
        //Reference a1 = Object & memory allocate
        a1.display();

    }
}
```

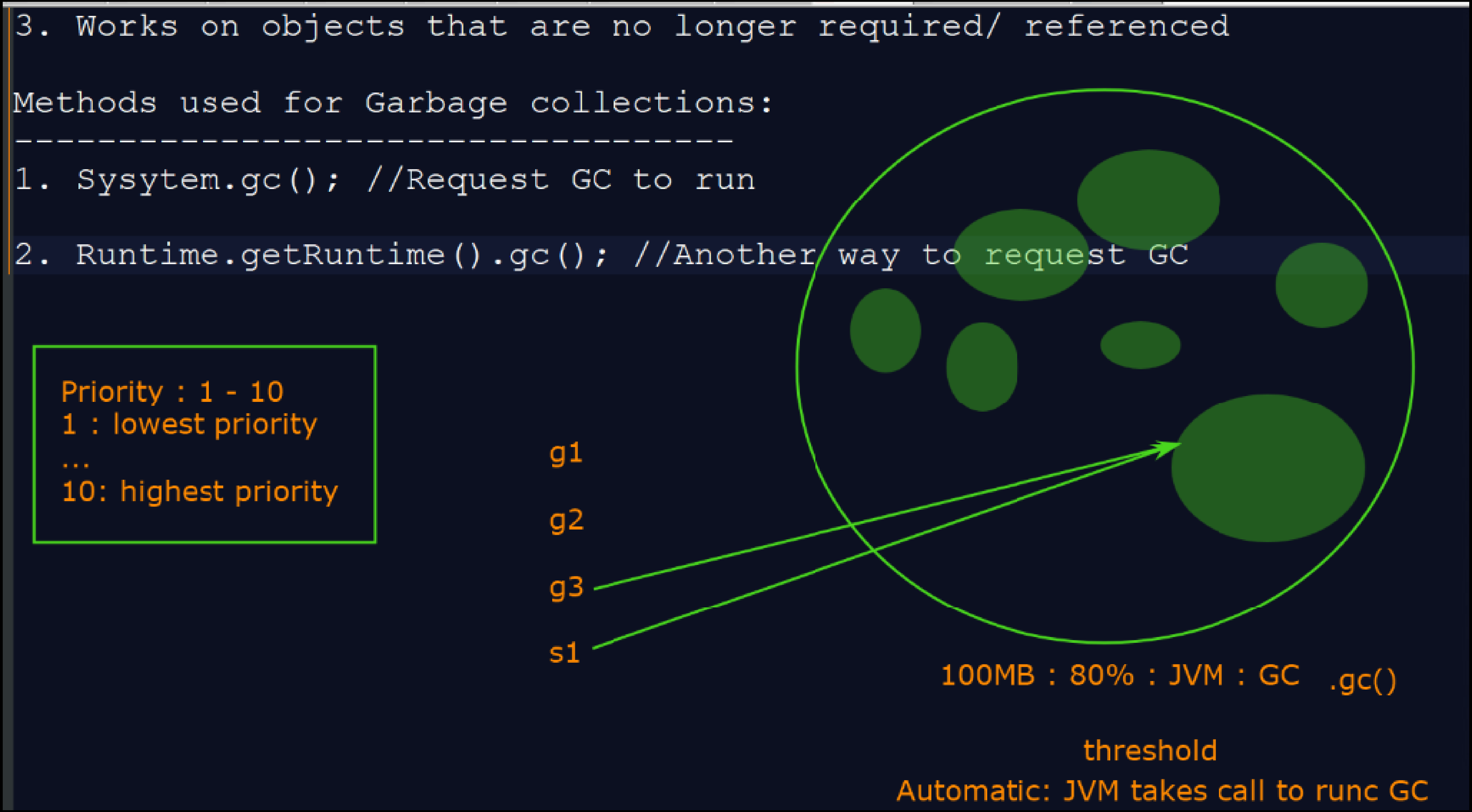
```
Key pointers:
1. Helps in automatic memory management
2. Uses the Garbage Collector (GC) to free up the heap memory
3. Works on objects that are no longer required/ referenced

Methods used for Garbage collections:
-----------------------------------
1. Sysytem.gc(); //Request GC to run

2. Runtime.getRuntime().gc();
```



100MB : 80% : JVM : GC .gc()

threshold

Automatic: JVM takes call to runc GC



3. Works on objects that are no longer required/ referenced

Methods used for Garbage collections:
----

1. Sysytem.gc(); //Request GC to run

2. Runtime.getRuntime().gc(); //Another way to request GC

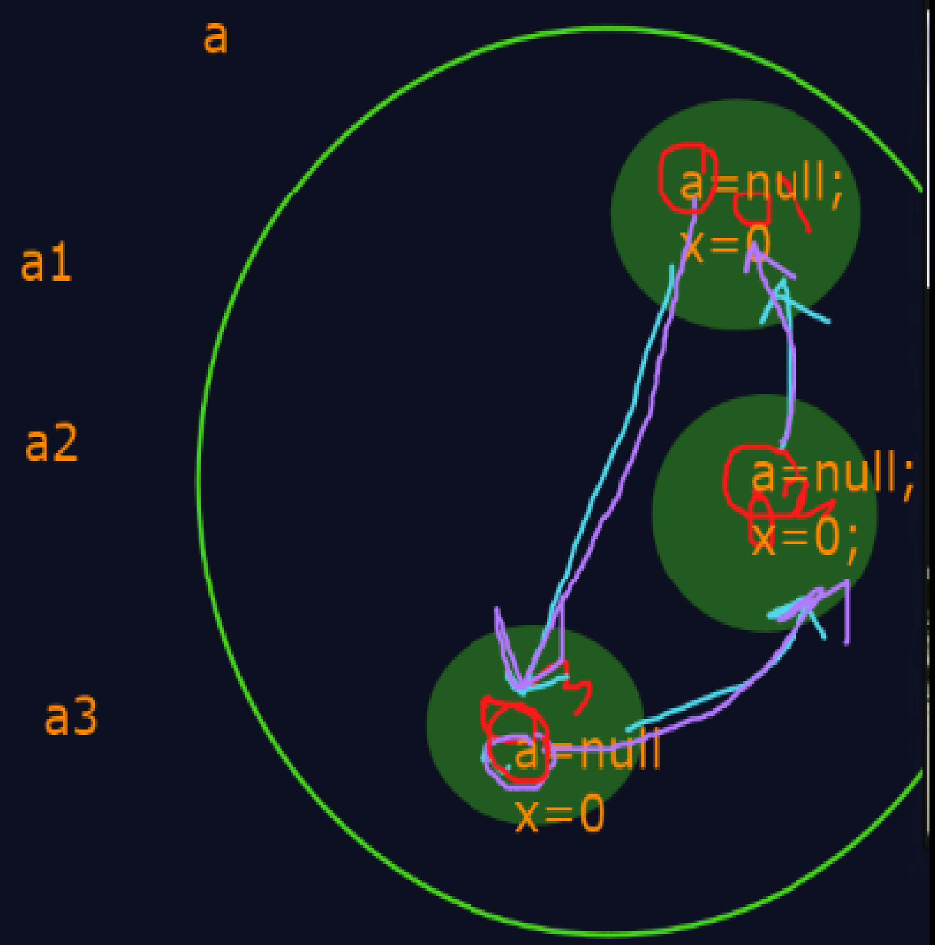
Priority : 1 - 10
1 : lowest priority
...
10: highest priority

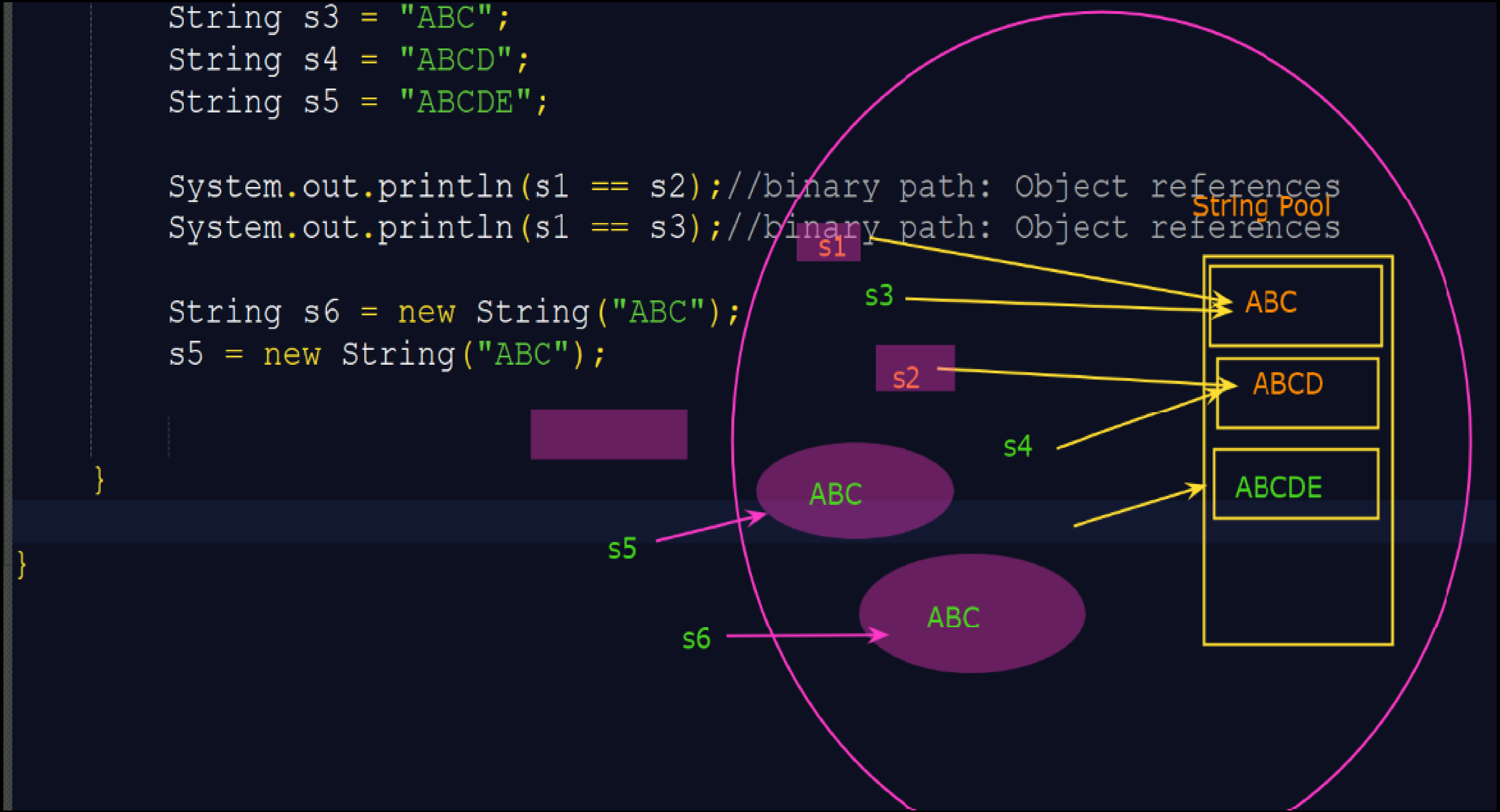
100MB : 80% : JVM : GC .gc()

threshold
Automatic: JVM takes call to runc GC

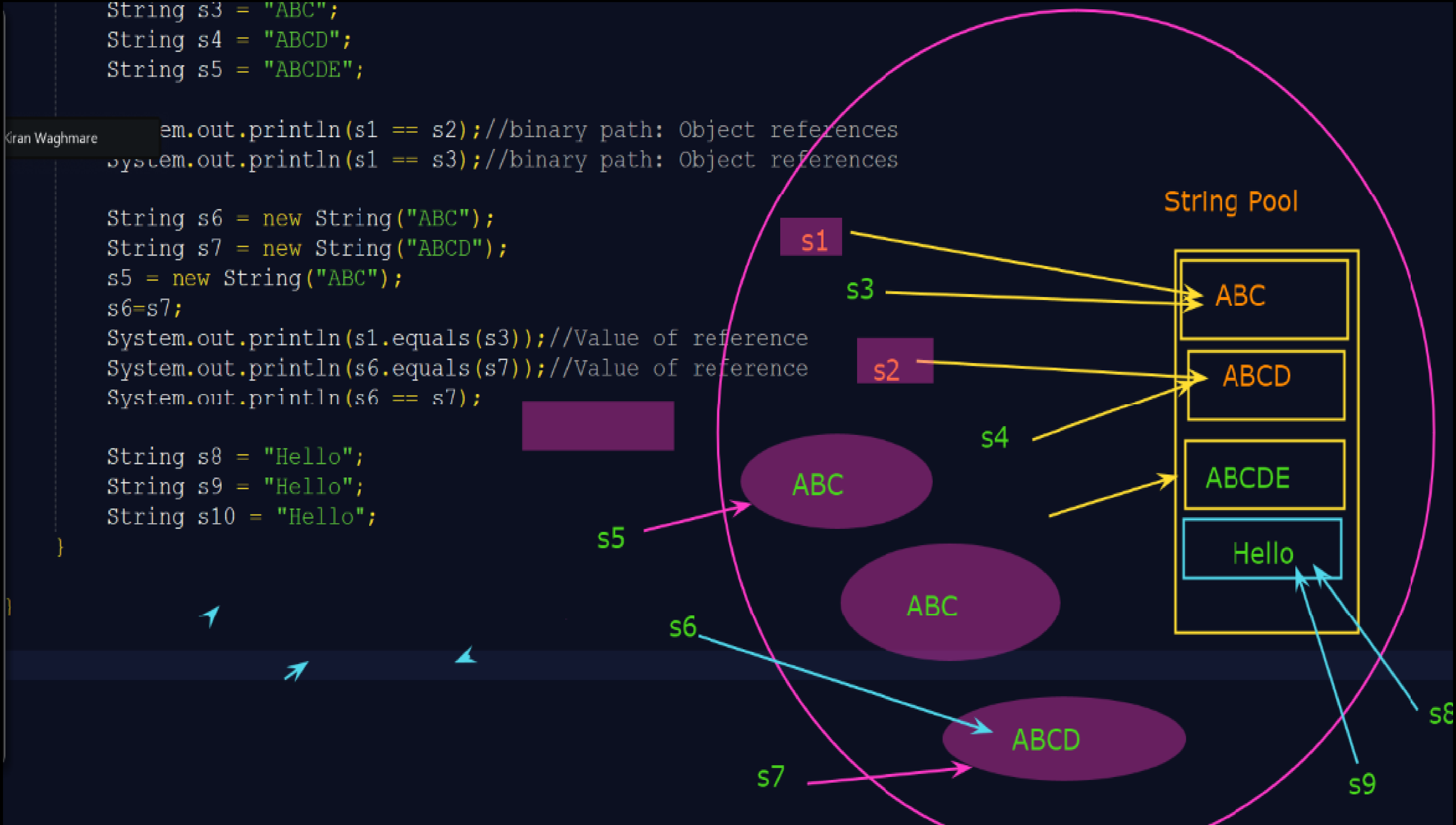
3. Island of Isolation:
-When object reference each other but are not reference elsewhere, they become unreachable

```
A a1 = new A();
A a2 = new A();
A a3 = new A();

a3 = a1;
a1 = a2;
a2 = a3;

class A{
    A a;//Reference
    int x;

    p.s.v.main(){

    A a1 = new A();
    A a2 = new A();
    A a3 = new A();

    a1.a = a3;
    a3.a = a2;
    a2.a = a1;
```

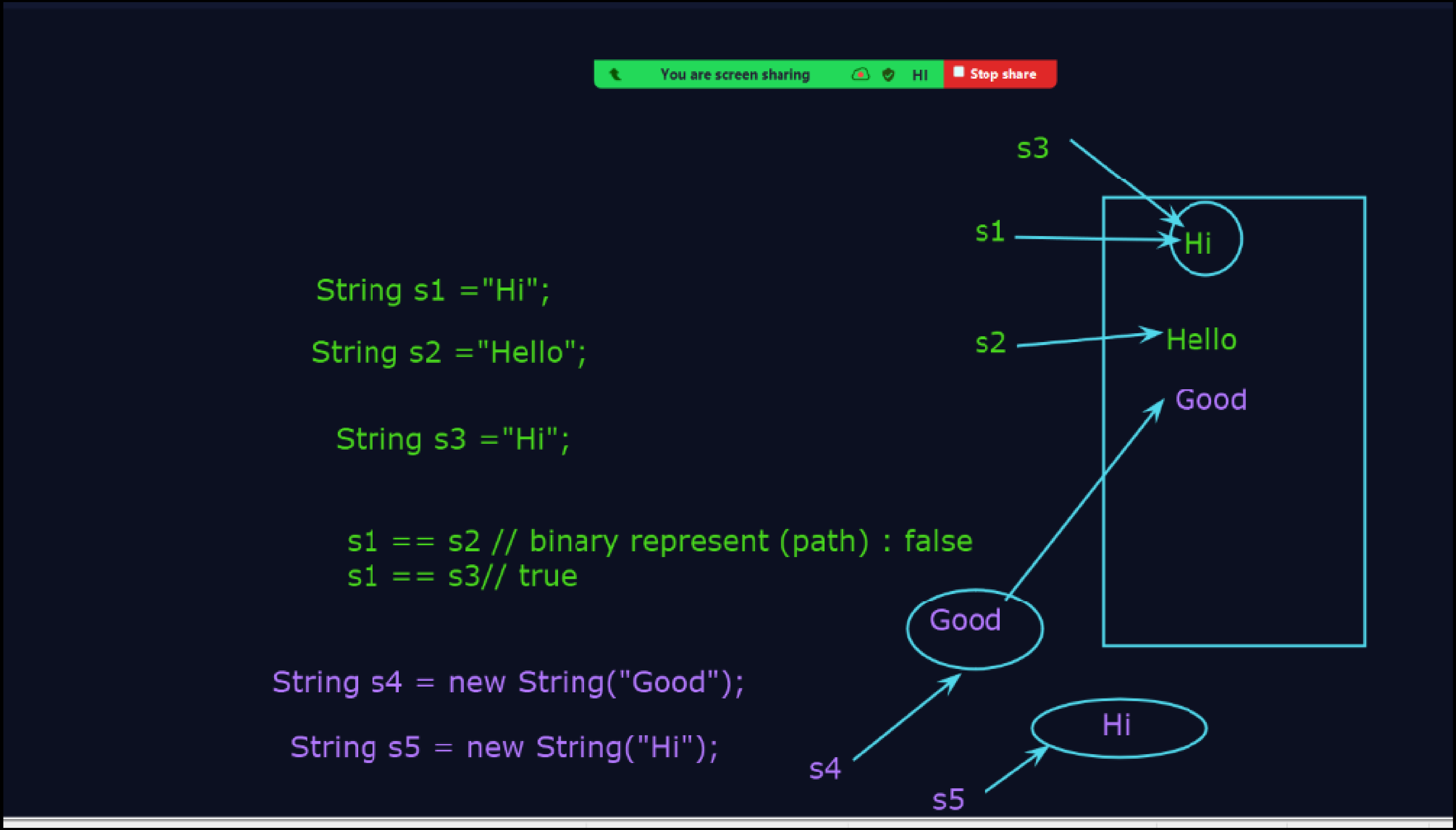




```
        String s3 = "ABC";
        String s4 = "ABCD";
        String s5 = "ABCDE";

        System.out.println(s1 == s2);//binary path: Object references
        System.out.println(s1 == s3);//binary path: Object references

        String s6 = new String("ABC");
        s5 = new String("ABC");

    }

}
```



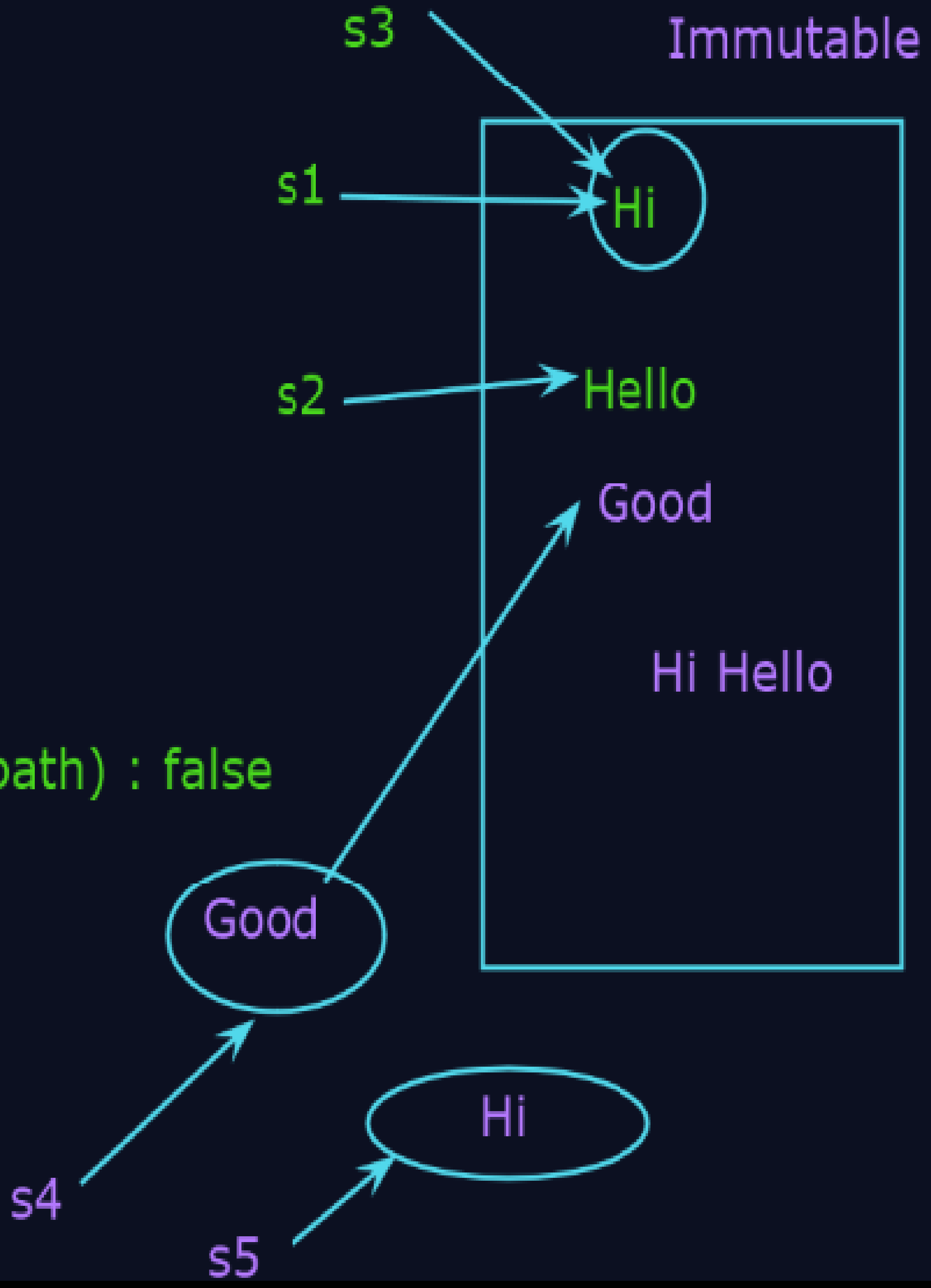
```
        String s3 = "ABC";
        String s4 = "ABCD";
        String s5 = "ABCDE";

              em.out.println(s1 == s2);//binary path: Object references
        System.out.println(s1 == s3);//binary path: Object references

        String s6 = new String("ABC");
        String s7 = new String("ABCD");
        s5 = new String("ABC");
        s6=s7;
        System.out.println(s1.equals(s3));//Value of reference
        System.out.println(s6.equals(s7));//Value of reference
        System.out.println(s6 == s7);

        String s8 = "Hello";
        String s9 = "Hello";
        String s10 = "Hello";
    }

}
```



```
String s1 ="Hi";

String s2 ="Hello";

String s3 ="Hi";


s1 == s2 // binary represent (path) : false
s1 == s3// true


String s4 = new String("Good");

String s5 = new String("Hi");
```

String

```
String s1 ="Hi";

String s2 ="Hello";

String s3 ="Hi";

s1 == s2 // binary represent (path) : false
s1 == s3// true

String s4 = new String("Good");

String s5 = new String("Hi");
```


s3
Immutable
s1
Hi
s2
Hello
Good
Hi Hello
Good
Hi
s4
s5