Array is an object in Java. All the objects which are created in Java are stored in a special area of memory called HEAP. The array is always stored in Heap irrespective of the elements of the array being primitives

A good rule of thumb is when the new keyword is involved the result will be on the heap. Only the reference variable will be stored in the stack and that variable will be pointing to the object stored in the heap memory

Everything is passed by value in java, there is no such thing as pass by reference in java. There are no pointers in java. There are no operators like \* or & in java like in C++

Every objects lives in heap, only the reference variable of those objects like in stack and point to the object in the heap

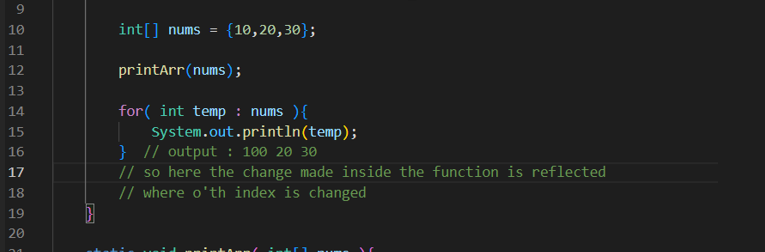
The heap objects, even the arrays that lives on heap memory may or may not be continuous

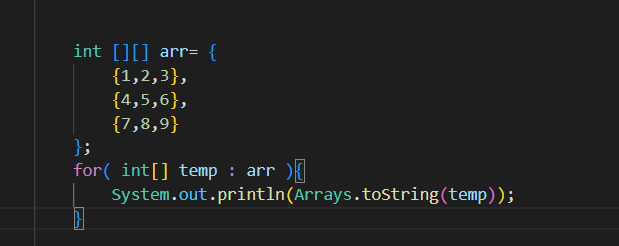
Null can only be assigned to non primitive data types, we can not assign null to any primitive data types. For example : int num = null; will gives us error

String str = null ; will not give any error

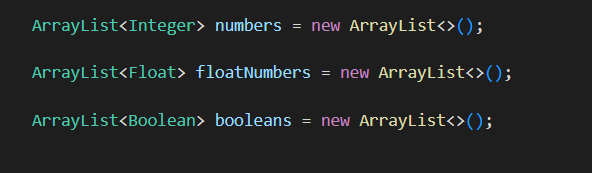
Null is the by default value of the reference variable, meaning if the reference variable is not initialized or assigned, then by default it points to the null

Here in java we have simpler for loop for arrays





reference variable in stack pointing to object living in heap

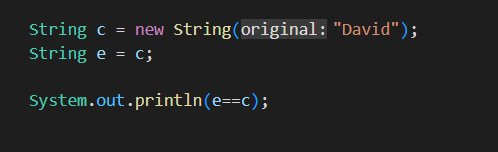


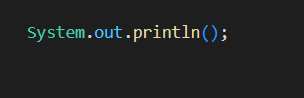
whenever new comes, the reference variable lives in stack, the object lives in heap and reference variable points to the object in the heap

Strings are immutable in java for security reasons

== will give true only when

1. both objects have the same value
2. both objects have the same reference variable



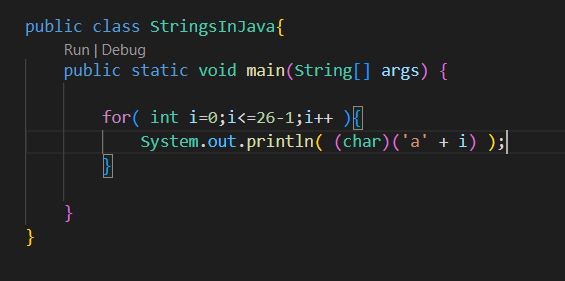


Here System is a class, out is a variable of type Printstream, and out has a method println( )

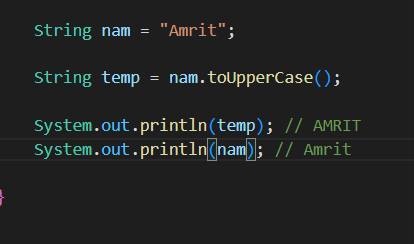
Here since out object is invoked from inside the System class directly, so out is an static member of the System class

A popular error / exception in java is **null pointer exception**

when we do something like null.toString( ), basically null.anything then we will get this type of error



On doing this all the English alphabets will be printed



Strings in java are immutable

We can not make the outer classes static, only the inner classes can be static, and inner classes are made static because we do not want them to depend upon the objects of the outer classes rather than on the outer classes, so it’s easy to make objects of inner classes and to work with them

Singleton class

**Conclusion**

If one object is not created than it will create it and return it

If one object is already created than it will return that

Child ko constructor bhitra surumaa super () lai call garnae and only assign the values to the variables of the child class, but the first line inside the child constructor is to call super( )

We can have reference variable of parent class pointing to the object of the child class.

**Only the properties of the data type of the reference variable will be accessed by the reference variable through dot operator AND NOT OTHERS, for example if the reference variable is of parent class type it will only access the properties of the parent class not the child class, except during the run time polymorphism by the method overriding**

We can access the properties of super class in sub class using super keyword instead of this, actually this is best practice since it let us know that we are accessing the properties of the super class

We can inherit as well as overload the static methods but we can not override the static methods, why this, might be asked in interview?

Interface only tells what must do not how to do

Creating separate classes for each interface is actually a good practice