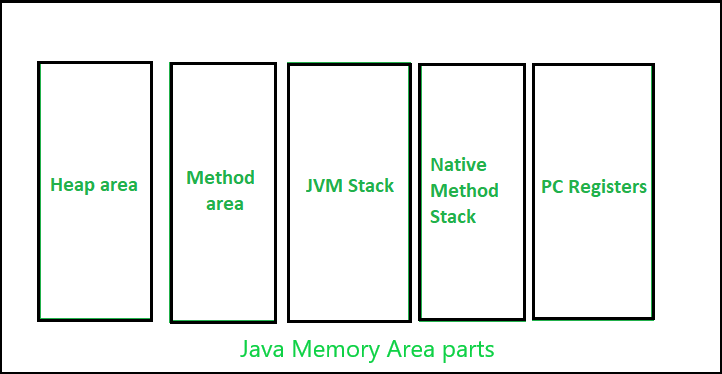
1. What is metaspace and heap memory?

Metaspace is a new memory space – starting from the Java 8 version; it has replaced the older Pegmen memory space. The most significant difference is how it handles memory allocation.

Specifically, this native memory region grows automatically by default

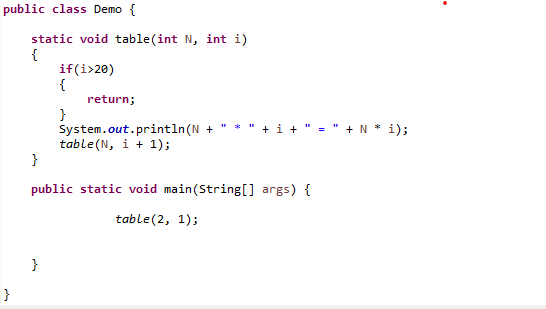
 JVM Memory Structure is divided into multiple memory area like heap area, stack area, method area, PC Registers etc. The following image illustrates the different memory areas in Java:

[](https://contribute.geeksforgeeks.org/wp-content/uploads/Memory.png)

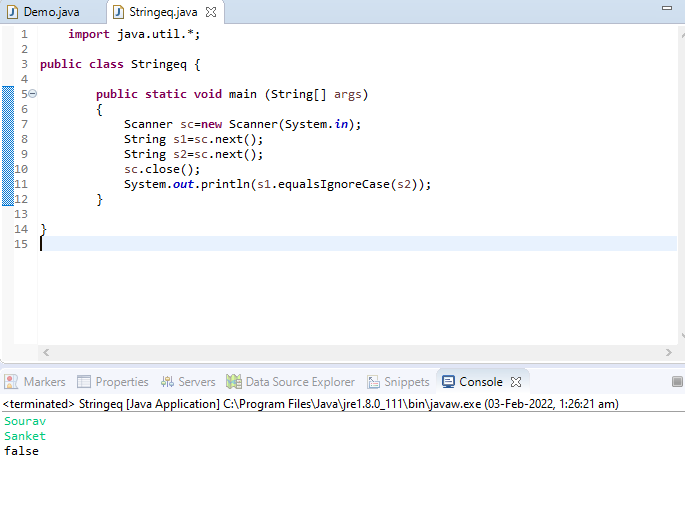
*JVM Memory area parts*

Here, the heap area is one of the most important memory areas of JVM. Here, all the [java objects](https://www.geeksforgeeks.org/classes-objects-java/) are stored. The heap is created when the JVM starts. The heap is generally divided into two parts. That is: 

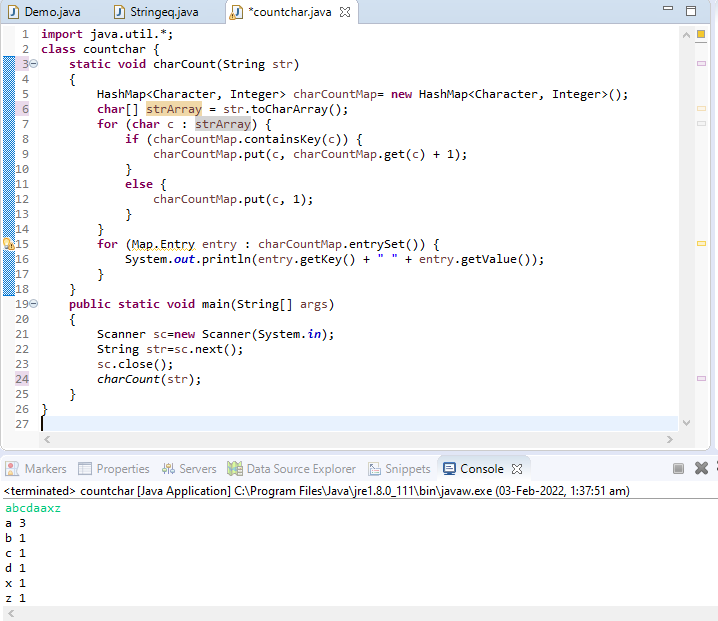
1. **Young Generation (Nursery):** All the new objects are allocated in this memory. Whenever this memory gets filled, the [garbage collection](https://www.geeksforgeeks.org/garbage-collection-java/) is performed. This is called as *Minor Garbage Collection*.
2. **Old Generation:** All the long lived objects which have survived many rounds of minor garbage collection is stored in this area. Whenever this memory gets filled, the garbage collection is performed. This is called as *Major Garbage Collection*.
3. Generate multiples of 2 until 20 using recursive function



1. Check if two strings are equal or not



1. Print the character count in a string.



1. Why java is platform independent?

Java compiler produces a unique type of code called bytecode unlike c compiler where compiler produces only natively executable code for a particular machine. When the Java program runs in a particular machine it is sent to java compiler, which converts this code into intermediate code called bytecode. This bytecode is sent to Java virtual machine (JVM) which resides in the RAM of any operating system. JVM recognizes the platform it is on and converts the bytecodes into native machine code. Hence java is called platform independent language.

1. Can we create class as final?

Yes, we can declare that your class is final; that is, that your class cannot be subclassed. There are (at least) two reasons why you might want to do this: security reasons and design reasons.

For example, if you wanted to declare your (perfect) ChessAlgorithm class as final, its declaration would look like this:

final class ChessAlgorithm {

. . .

}

Any subsequent attempts to subclass ChessAlgorithm will result in a compiler error such as the following:

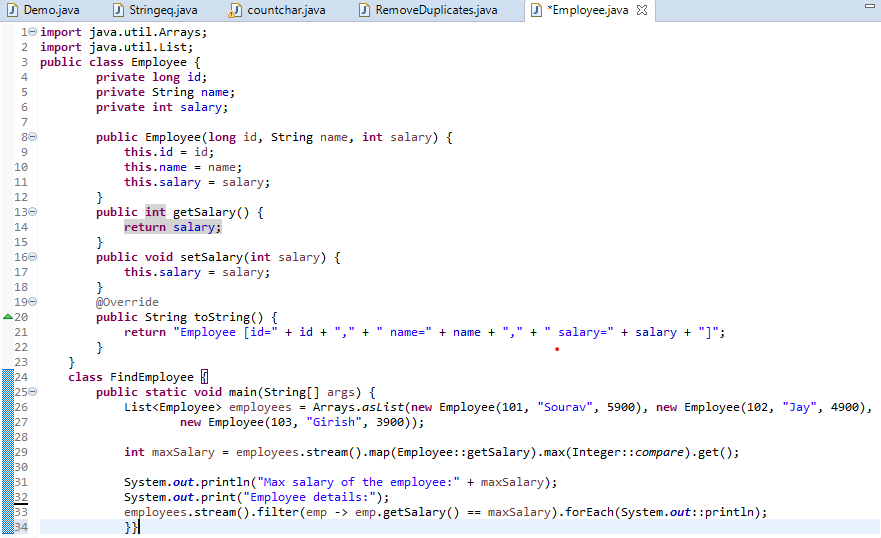
Chess.java:6: Can't subclass final classes: class ChessAlgorithm

class BetterChessAlgorithm extends ChessAlgorithm {

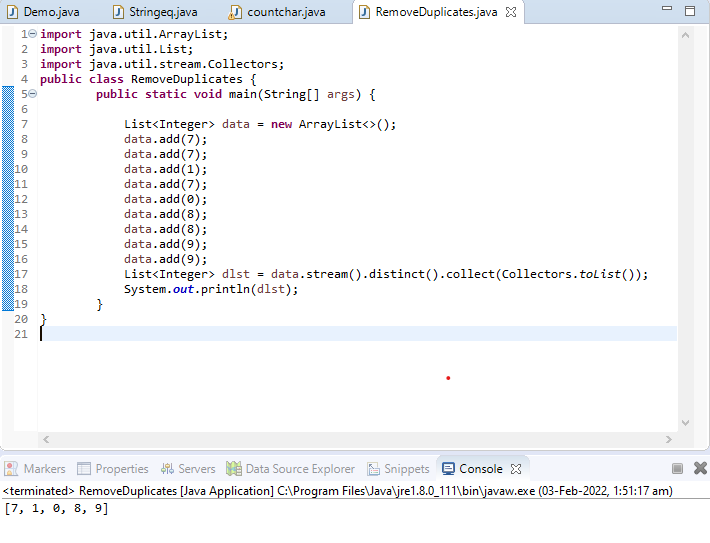
^

1 error

1. Consider we have employee class with empid, empname and salary and list of employees get the highest salary paid employee data



1. Consider a list of duplicate values remove duplicate value and get unique values from the list



1. Can we write try and finally without catch block?

Yes, we can have try without catch block by using finally block.

We can use try with finally, as we know finally block always executes even if you have exception or return statement in try block except in case of System.exit()

.

1. Create a java application for College Management.

