Sunday, May 1, 2022 8:09 AM

Garbage Collector --- Implicit Memory Management

Implicit = internally the JVM will manage the memory instead of the programmer.

Manage which memory ? Heap in the JVM in the RAM
JVM sections

When we talk of memory management
JVM = the Java process

When we talk of compilers or interpreters
JVM = interpreter ---- java command used to run the application

Java Process has Sections

Class Area	Stack Area	Heap Area
All the .class files are loaded here	Thread stacks that hold method calls and local variables	All the objects

All the objects get space in the heap area !!!
The space must be released when objects are no longer used.

Heap Space is FINITE !!!! SO if not properly managed then the heap will be full , when the heap is full program crashes with a JVM Error that occurs at runtime .java.lang.OutOfMemoryError

---

Syntax Error ----- compile time error ----- class file is not created
JVM Error ----------run time error --------it occurs in between program execution and program ALWAYS crashes
Exception ----------run time problem ----it occurs in between program execution and program MAY CRASH or the Exception MAY be handled .

Garbage Collector --------- thread of the jvm that will run periodically
It will trace the objects for REACHABILITY from the stack
It will **mark** all the UNREACHABLE objects
The GC calls a method of Object class --- **finalize**()
It will **sweep** all the marked objects } release all the objects

mark and sweep GC !!

Object class has a finalize() method . It is empty default implementation .
This method is called by GC after marking before sweeping !!
If we override the finalize we can see that the GC is about to release that object

---

Good Programming Practices for GC to be able to mark and sweep ----------

As far as possible --- keep the scope of the reference variables as low or less as possible
So that the objects quickly become unreachable and GC may release them

Whenever you are finished working with a reference make it **null** !!! -- the object it is pointing to may become unreachable and available for GC .

_____

How to explicitly call a GC ? **System.gc()**

Syntax Error ----- compile time error ----- class file is not created

JVM Error ----------run time error --------it occurs in between program execution and program ALWAYS crashes } OutOfMemoryError for Heap , StackOverFlowError for Stack

Exception ----------run time problem ----it occurs in between program execution and program MAY CRASH or the Exception MAY be **handled** .

Exception Handling ----------

Avoid program crashing if exception occurs at run time ----
To avoid this -- use **try - catch**