What is the use of final, finally and why there is no multiple inheritance in Java?

**Exception Handling:**

* Try, catch, finally, throw, throws
* Basic way of handling the exception should be centralized (centrally handled or globally handled).
* To handle the exception centrally we use the keyword throws.

Types:

* System defined Exception
* User Defined Exception

Throw is used to create user defined exceptions.

**Throwable**

* Super class of Exception Class and Error Class
* Super class of all the Exceptions are Exception Class.
* We can have Error Class if there is possibility of System Error .

Throwable

/ \

Exception Error

Checked Exceptions:

* Any resources used must be handled compulsorily, without exception being handled the code won’t be compiled

Unchecked Exceptions:

* Any resources used need not be handled, which can be handled, but still the code will be compiled
* Ex: Array index, NullPointer.

**User Defined Exception:**

* All the business rules are User defined Exceptions, these business rules are taken from the client/user
* Any user defined exceptions is create by extending a **Exception Super Class.**
* If any class extends Exception Super Class, it becomes a exception
* System cannot handle the user defined exception, it should be handled by throw statement.

**Finally**:

Purpose of Finally?

Before executing break, continue and return, finally block is executed.

**Generics**

* Is a template class that is created without datatype.
* Datatype is given at the time of object creation, so that same class can be used for different datatypes, instead of writing multiple classes.
* In generics the datatype should be always Wrapper classes or Classes, It doesn’t support primitive types.
* Generic templates should be used for storage rather than computations.
* Generic interfaces can be used for computation. **Note: It should be a functional programming.**

------------------------------------------------------------------------------------------

**Collections**

* Either collections or Arrays is to store the data before or after processing. Once the processing is done we can remove the data from the collections.
* Collections are dynamic, Arrays are static.
* In java collections is a framework(contains Api, interfaces ,algorithms, classes) which is going to do the job on behalf of us.
* In collection framework we have got 2 main interfaces,
  + Collection -Only values
    - Under collection we have got
      * List Interface: ArrayList(asynchronous), Vector(synchronous), LinkedList
        + Can be accessed by the index
      * Set: HashSet(unordered), TreeSet(sorted), LinkedHashSet(ordered)
        + Cannot be accessed by the index
      * Dequeue: ArrayDequeue, LinkedList
      * Queue :LinkedList
  + Map- (key, value) pair
    - HashMap(key->unordered)
    - TreeMap(key-> sorted order)
    - LinkedHashMap(key-> ordered)
* Collections class is a algorithm.