__/__/____

String :-

* In java string is basically an object that represent sequence of char values.
* java String provides a lot of methods to perform operations on String such as compare(), concat(), equals(), split(), length(), replace() compareTo(), substring() etc.

ways to create a String object

There are two ways to create String object

1) By string literal
2) By new keyword

1) By String literal :-

If we create a string by using string literal, each time JVM checks the string constant pool first if the string already exist in the pool the reference of already existing instance is returned, if the string doesn't exist in the pool a new string instance is created.

Ex String s = "Rohini";

Ex for new keyword :-

String s = new String("Rohini");

Adv of string literal :-

To make java more memory efficient we use string literals.

2) By using new keyword :-

*The string can also be declared using new operator i.e dynamically allocated. In the case of strings are dynamically allocated they are assigned a new memory location in heap

*This string will not be added to string Constant pool, If you want to store this string in the Constant pool then you will need to "intern" it.

Note :- *Why string is immutable

Why because Once the string object is created it's cannot be changed but a new string object is Created

* The reason behind the string Class is being final is because no one class override the methods of the string class

* Reasons behind string is immutable

-> classloader :- A classloader in java uses a String object as an argument

__/__/____

→ Thread safe, security, Heap space

Java StringBuffer Class :=

* Java StringBuffer class is used to create mutable string objects. The StringBuffer class in java is the same as string class except it is mutable, i.e it can be changed.
* It has some inbuilt methods & constructor,

Constructor :- StringBuffer()
StringBuffer(String str)

Methods :- append(), replace(), delete, reverse(), length()

*Note :- A string that can be modified & changed without creating new object is known as mutable string, StringBuffer & StringBuilder classes are used for creating mutable string.

Java StringBuilder Class :=

* Java StringBuilder class used to create mutable String, The java StringBuilder is same as StringBuffer, except that is not syncronized.

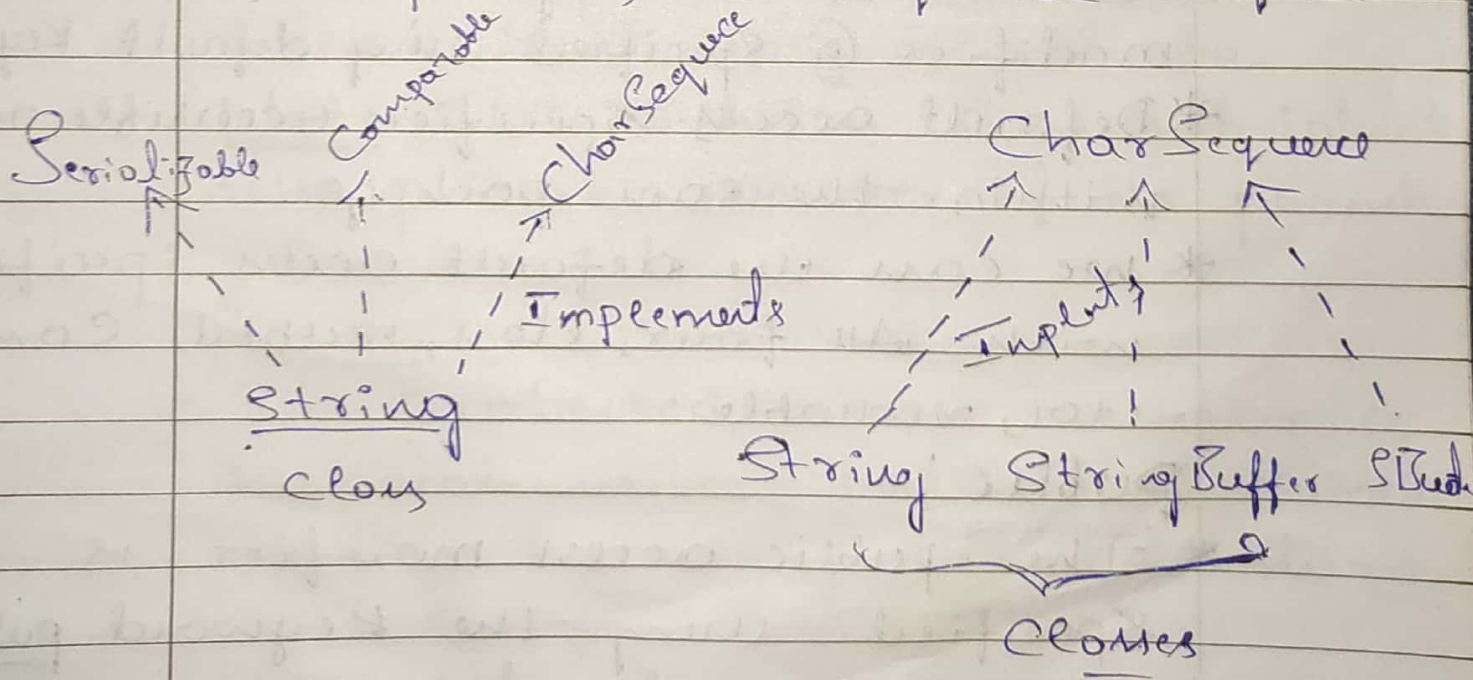
__/__/____

* intrueld constructor; (— same as above)
* intreil methods length(), substring()
  charAt().

difference between StringBuffer & Builder

StringBuffer	StringBuilders
* StringBuffer is Syncroni-zed i.e thread safe. means two thread can't call the method of StringBuffer simultaneaky	* non-syncronized. i.e not thread safe means two thread can call the methods of SB simultaneously
* less efficient	* more effient
* introduce in java 1.0	* java 1.5

Note := String class implements Seriolizable, Comparable, CharSequence interface



__/__/____

Access Specifier :-

As - the name suggest acces modifier (specifier) in java helps to restrict the scope of a class, Constructor, variable, method,

There are four types of acces modifiers (specifier) in java.

1> Default
2> Private
3> protected
4> public

Default :- when there is no access specifier for a class, method, It is said to be having the default access modifier @ specified using default keyword

* Default access specifier accessible only within the same package
* we can use default access specifier with All four, class, method, Constructor, variable

Public :-

* The public access modifier is specified using the keyword public
* class, methods, data menembers

__/__/____

-That are declared as public are accesible anywhere in the program.

Private :=

* private access modifier is specified using the keyword private.
* The methods & data members decla-red as private then it accesible only within the class
* If you make any contructor as private, than you can not create instance of that class outside, But we can declare the constructor as private
* we cannot declare a toplevel class as private, But we can declare inner class as private.

Note := A subclass does not inherit the private member of its parent class.

Protected :=

* The protected access modifier is accessible within package & outside package but only through inherit-ance only.

__/__/____

* The protected class modifier can be applied on the data members, method & Constructor, It can't be applied on the class, though It can be applied to inner classes.
* there the subclass inherits the members of superclass, But It is not accessable

Note :- Even though subclass inherits the members of superclass But It is not accessable outside the package, Inside the package it is accessable, But the accessibility of outside the package is to override,

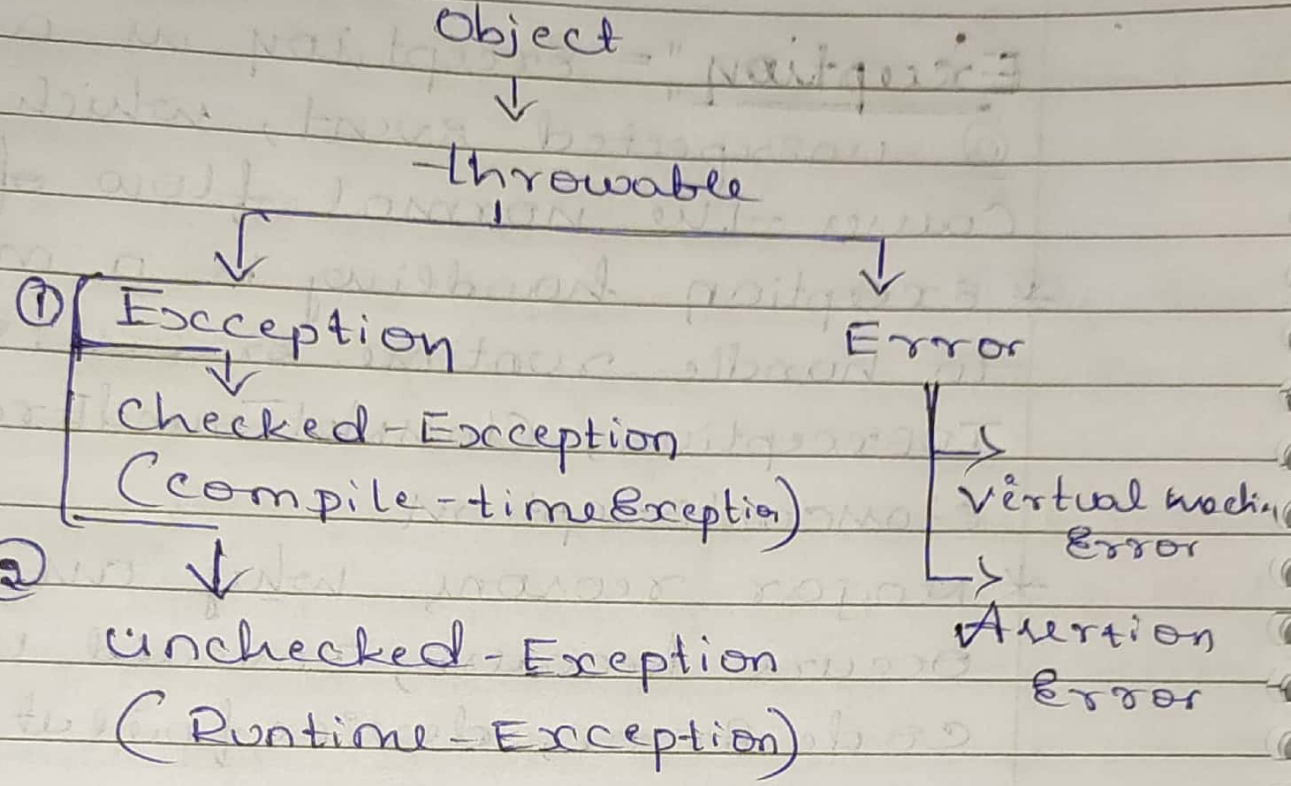
__/__/____

Exception handling :=

Exception := Exception is a unwanted @ unexpected event, which may causes the normal flow of program

* Exception handling is a mechanism to handle runtime errors, such as IOException, ClassNotFoundException, CloneNotSupportedException
* Major reasons why an Exception occurs because, Invalid user input, code error, device failure, loss of network connections
* the difference between Errors & Exceptions, Errors indicate a serious problem that a reasonable application should not try to catch. where Exception indicates condition that a reasonable application might try to catch.
* Throwable is the base class of the exception hierarchy
* But we we know Object is the parent most for all class in java

Exception hierarchy :-



Types of exceptions

1) Built-in Exception

* **Checked Exception** :- Checked Exceptions are called compile time Exception, because these Exceptions are checked at compile-time by compiler, that is why compiler forcing us to handle, data issue, recor
* **Unchecke Exception** :- Compiler not forcing us to handle is known as unchecked

1) JDK, JVM, JRE -> Refer howtodoinjava.com. & Compeete reference.

2) Data type -> Javatpoint, w3school, CR.

3) Java Buzzword & Java feature (Compeete Ref)

Note !=

* Array & class in java are always mutable, in that you can assign new value to elements of Arrays.
* Transient keyword in java transient is a variable modifier, used in serialization At the time of serialization, if we dont want to save value of perticular variable in a file, then we use transient keyword then JVM ignore original value of variable & save default value.
* boolean to string conversion
  > string.valueof(boolean)
* Stack-trace (Heirachy of method where the exception occur.
* Seriolization != Converting an object into Byte Stream.
* Deseriolization != Converting byte stream into an object.

__/__/____

Exception, It's all code issue & logic issues, which can not be recoverable.
Ex NullpoiterException, ArrayIndexoutofBond

Event Creation :-
We can create event by 'throw' keyword, with the reference of throwable, Exception, and By using reference of Eception's subclass, Aswell as the refecence of Error, Eventhrough, by using the refernce of Throwable, & error is not recoverable, just because, i.e too generic & It also may reduces the stack race visibility.

Methods of Throwable class :- public void
getCause(), getMessage(), printStack
Trace() throwable, toString() string, getStackTrace()

Ways to handling the Exception

1) try and catch
2) deligate

1) try & catch block

__/__/____

The java try block is used to enclose the code that might throw an exception. & java try block must be followed by either catch & finally block.

Java catch-block :- Java

* Catch-block is used to handle the exception by declaring the type of exception within the parameter the declared exception can be parent class exception & generated exception type. However, the good approach is to declare the generated type of exception.
* The catch must be used after try catch only, we can use multiple catch block with a single try block. At a time only one exception occurs & at a time only catch block is Executed.
* Finally is block which is always execute, whether an exception is handling & not, therefore it contains all the necessary statements that needs to be printed regardless of

__/__/____

the Exception occurs @ not.

* For each try block there can be zero @ more catch blocks but only one finally block.

* we can stop the execution of finally block by calling System.exit(), that will terminate the JVM

Advantage of Exception Handling

The Core Adv of exception handling is to maintain the normal flow of program. An exception normally disrupts the Normal flow of application that is why we need to handle Exception, if we do so we can continue the normal flow.

By Delegate :=

By using throws keyword we can delegate declare an Exception, It gives an information to the programmer that there may occur an Exception, so it is better for the programer to provide Exception handling code so normal flow of program can be maintained.

__/__/____

What happens if Exception is not handled

* firstly JVM looks for the method where Exception has been occurs
* if the Exception is not handle by the particular methods, JVM looks into methods which has been called that particular method
* Then JVM force to main method for handling default Exception, that perform following task
  > print the stacktrace
  > terminate the program.

__/__/____

Polymorphism :=

In java refers to objects capacity to take several forms

2 types :=

1> Compile-time polymorphism

2> Runtime-time polymorphism

Compile-time polymorphism := which is also known as static polymorphism & early binding, compile-time polym-orism is a polymorphism that is resolved during the compilation pro-cess. Overloading of methods is is called through the reference varaible of class. Compile-time poly morphism is achived by the metho-d overloading.

Run-time polymorphism

When ever an object is found with the fuctionality at runtime, @ JVM Checks wether the refe-rence which has to be take from Super class & sub class, Runt-time polymorphs (Dynamic.) come to pic-ure when we are deding with

__/__/____

Abraction :=

Abstraction is a process of hiding the implementation details & showing only functionality to the user
Real-word-Ex :- Sending SMS, where you type the text & send the message. you don't know the internal processing about the message delivery.

ways to achieve Abstraction :=

There are two ways to achieve abstraction in java

1) Abroact class :- we may partial Abstract
2) Interface :- Compleate Abstraction.

1) Abstract class :-

A class which is declared as abstract is known as an abstrad class. It have both abstract & non-abstract methods. Abstract class needs to be Extended why because the implemention of abstract method that has to be provide in sub-class only, so you doesn't want to provide implemention in subclass they sub class also has to be declare with Abstract keyword.

__/__/____

Note:- *Abstract class cannot be instantiate.
* Abstract class have Constructor & static method, Also
* we can provide impelmention for non-abstract method in same class.

Ex

```
abstract class Bike{
    abstract void run();
}

class Honda Extends Bike{
  void run()
  {
  System.out.println("running safely");
  }
public static void main(String args)
      Bike obj = new Honda();
          obj.run();
      }
    }
```

* If - there is an abstract method in class, - that class must be abstract.

__/__/____

Encapsulation :=

Encapsulation in java is a process of wrapping code & data together into single unit.

real-word-Ex :- A capsule which is mixed of several medicines.

We can achieve Encapsulation By making all the data members of the class as private, If we that respective class declared with private mean that can be accessible with in the same class. So we have to getters & setters methods to get & set the values of the private fields.

Advantage of Encapsulation

* It is a way to achieve data hiding in java because other class will not able to access the data through the private data members.
* The Encapsulate class is easy to test. so that it is better for unit testing
* It provides controll over data, i.e Suppose you want to set the value

__/__/____

of id which should be greater than 100 only, then can write the logic inside the setter method.

* these closes provides security.

Ex

```
public class Studut {
    private String college = "AKG"
public String getcollege() {
    return college;
  }
}
```

Interface :-

Interface in java is nothing but blueprint of a class. It helps two more classes interact with each other. We can achive interface by using 'implements' keyword.

In other words, you can say that interface can have abstract methods, since java8, we can default and static methods in an interface, since java 9, we can have private method & variables must be public, static & final, & By default all methods in interface is public & abstract.

Note :=

* Interface cannot be instantiated just like the abstract class

Advantages of Interface :=

* Interface is used to achieve abstraction
* It is used to achieve multiple inheritance in java
* It can be used to achieve loose coupling.

Declare an interface :- An interface is declare by using the 'interface' keyword

__/__/____

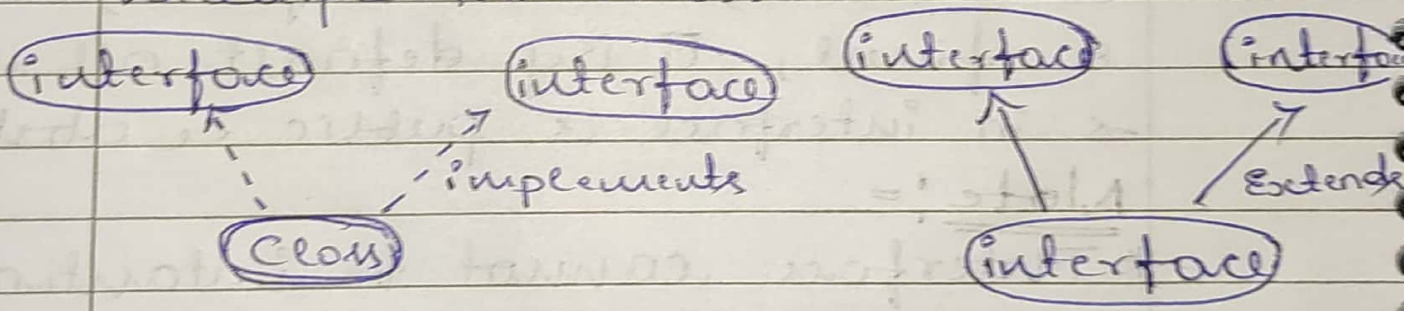
& achieve interface by using implements Keyword.

Relationship between class & Interface

A class extends another class, an Interface extends another interface But a class implements an Interface.

Multiple inheritance in java By Interface

If a class implements multiple interface, & an interface extends multi-ple interface, It is known as multiple inheritance.



Multiple Inheritance

```
interface printable{
  void print()
}
interfer Showable{
  void show()
}
}
class A7 implements printable, Showable
{
```

```
public void print()
    {
    System.out.println("Hello");
    }
public void show()
    {
    system.out.println("welcome");
    }
public static void main(String[] args){
  A7 obj = new A7()
    obj.print();
    obj.show();
    }
    }
```

Note:- use of default method which allow the interface to have methods with impelementation without affecting the clases that implement the interface.

Typy of Interface

1) Functional Interface
2) Marker @ taged Interface.

__/__/____

* Functional Interface :-
Functional Interface is an interface that has only pure one abstract method. It can have any no of static & default methods.
Ex :- Runnable, Comparator, Comparable, predicate.
Adv Note :- By using Functional Interface we can achieve 'lambda Expression'

* Marker Interface :- @ tagging Interf
An Interface that does not contain any methods is called MarkerInterfa
Ex :- Serializable, Cloneable
Adv :- It provides run-time type information about the objects so that compiler, & JVM have additional information about the object.

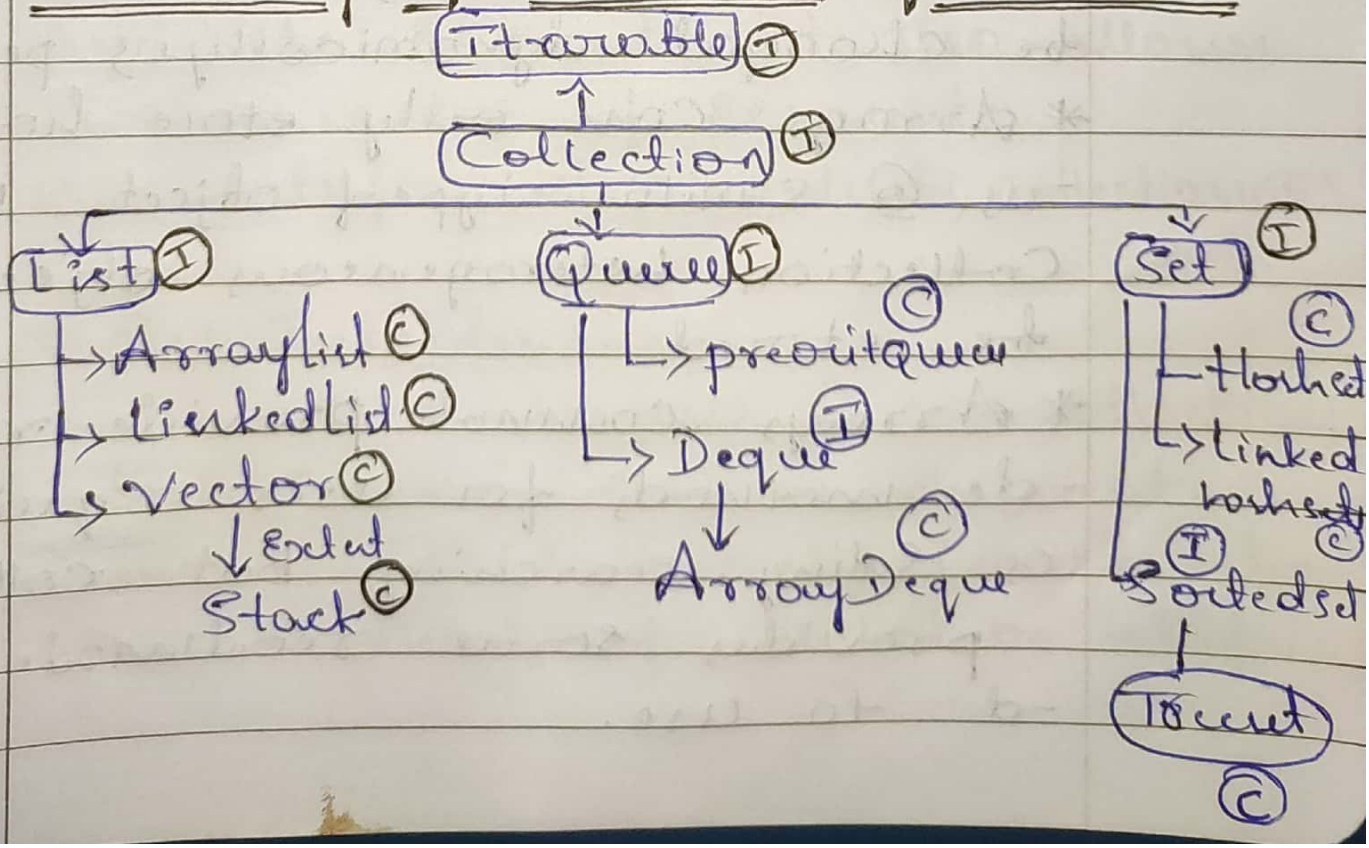
__/__/____

Collections :-

Java collections means a single unit of objects. ⓐ it is combination of interfaces & classes. collection is framework in java why because It provide readymade architecture, & It represents a set of classes & interfaces.

=> Collection framework is a combination of classes & interface which used to store & manipulate the data in the form of objects. It provides various classes such as Arraylist, Vector, Stack etc interfaces such list, set, queue. etc.

Hirarchy of collection framework



__/__/____

Methods of collection interface :-

There are many methods declared in the collection interface add(), addAll(), remove(), removeAll(), size(), clear(), contains(), containsAll()

diff b/w array & collection

Array & collections are somewhat similar to storing & manupulating the data, but they differ in many ways,

* Arrays are always of fixed size. i.e user can not increase & decrease the length of array at run-time, But in collections size can be changed dynamically as per need
* Arrays can only store homogeneous & similar type of object, but in Collection heterogeneous objects can be stored
* Arrays cannot provide readymade methods for user requirment as sorting, searching but collection provides some readmade method to use.

__/__/____

Various interfaces used in CF :=

As we know, collection framework implements various interfaces

1) Collection interface :

Collection (java.util.Collection) is the primary interface, rest of all interfaces are Extended by 'Collection' interface only.

Syn :- public interface Collection<E> Extends Iterable.

2) List Interface :=

List interface Extends the Collection interface, & It is an ordered collection of objects. It contains duplicate elements, It also allows random access of elements.

Syn :- interface List<E> Extends Collection<E>

3) Set Interface :=

Set interface is a collection which can not contain duplicate elements. It can only include inherited methods of Collection interface

Queue interface :-

Queue interface extends collection & it defines queue data str which stores the element in the form of First in first out (FIFO).

Deque interface :-

It is double ended queue It allows the insertion & removal of elements from both ends. It implants the properties of both stack (LIFO) & queue (FIFO).

Map interface :-

A map represents a key, value pair storage of elements, map interface doesn't implement the collection interface It can only contain unique key but can have duplicate element, there are two interface which implements the map interface that are map interface & Sorted map.

__/__/____

diff between Arraylist & vector

Arraylist	vector
* Arraylist not synchronized	* vector is Synchronized.
* It is not thread safe	* thread safe
* Arraylist increases its size by 50% of the array size	* vector increases its size by doubling the array size.

diff between Arraylist & linkedlist

Arraylist	Linkedlist
* the internal element of the arraylist is 'array'	* the internal element of linked list is node
* Arraylist is not efficient for manupulation	* Better to manupulate data.
* It is efficient to store & fetch data	* It is not much efficient as compare to arraylist
* It is provide random access	* It doesn't provid random Access
* It takes less memory It stores only object	* it takes more memory it stores object as well as Address of that object.

__/__/____

difference between Iterator & ListIterator

Iterator	ListIterator
* The Iterator traverse the elements in the forword direction only	* in both forword and backword direction
* The Iterator can be used in list, set, & Queue	* ListIterator can be used in List only.
* The Iterator can only perform remove operation while traversing the collection	* It perform add & remove operation

difference between Iterator & Enumeration

Iterator	Enumeration
* It is fail-fast	* It is not fail fast
* It is slower than Enumeration	* It is faster than iterator
* It perform remove operation while traversing the collection	* It perform only traverse operation on the collection

difference between list & set

the list & set both extend the Collection interface, however there are some difference between

them as well,

* List can contain duplicate elements whereas set includes unique items
* The list is an ordered collection which maintain the insertion order where as set is ordered collection which does not preserve the insertion order.
* The list contain single legacy class i.e vector, the set does not have any.
* The list interface allows 'n' number of 'null' value, whereas set interface only allows single null value.

difference between {HashMap & TreeMap} {HashSet & TreeSet}

Both are classes, which implement set interface

* HashSet maintain no order, whereas TreeSet maintains ascending order
* HashSet implemented by HashTable, whereas TreeSet implemented by Tree structure.
* HashSet perform faster than TreeSet.
* HashSet is backed by HashMap, where, TreeSet is backed by TreeMap.

difference between set & map

> Set contain values only, map containg keys & values both.
> Set contain unique value, but map contains unique key with duplicate value.
> Set holds a single nor of null value where as map can include a single null key with 'n' nor of null value.

diff b/w HashSet & HashMap

> HashSet contains only values whereas HashMap includes the Entry(key, value).
> HashSet can be iterated, but HashMap needs to convert into set to be iterated.
> HashSet cannot have duplicate value, HashMap can contains dupli-cate values with unique keys.
> HashSet contains the only single nor of null value, HashMap can hold a single null key with 'n' nor of null values.

__/__/____

HashMap	HashTable
* It is not synchronized	* Synchronized
* It contain one null key & multiple null values	* It cannot contain null key & null value
* non-thread safe	* Thread safe
* inherity the Abstract Map class	* HashTable inher-its the Dictionary class

difference between Collection & Collections

* The Collection is interface where Collections is a class
* The Collection interface provide the standard fuctionality of data structure to list, set, Queue, however Collections class is to sort & Synchro-nize the collection elements.
* Collection interface can provide the methods that can be used for data structure, where as Collections class provide the static methods which can be used for various operation on the collection.

__/__/____

difference between Comparable & Comparator.

Comparable	Comparator
* If we implement the Comparable interface the actual class is modified	* Actual class is not modified
* Comparable provides only one sort of sequence	* It provides multiple sort of sequence
* It provides one method called CompareTo()	* It provides one method Called Compare()
* It is found in java.lang package	* java.util

BlockingQueue:=

Blocking Queue is an interface which extends the Queue interface It provides Concurrency in the operation like retrieval, insertion, deletion, it cannot contain null elements.

Adv of propertyFile:=

If you change the value

in the properties file don't need to recompile the java class, so its makes the application load easy to manage. It is used to store the information which is to be changed frequently.

Adv Overrid Equals() method

Equals method used to check whether two objects are the same or not. It needs to be overriden if we want to check the objects based on the property.

Array	Arraylist
* Array is fixed Size	* we can size the size dynamicly
* Array can store primitive datatypes as well as objects	* It store only objects.