Shortcuts & Tips:

* HTML - To format the file - COMMAND + SHIFT + F
* XML - How to Comment - <!- - <xml tags> - - !>

Java Tutorials

**Topic 1 : History of Java?**

* Developed by James Hosling in 1991 but official is 1995 (when Java was named) while working in Sun Microsoft
* James Hosling was part of Green Team and it was developed for Television Interactive Programs
* Initially it was named as GreenTalk then Oak and after that Java was named finally after a team meeting discussion
* Since then Java has launched various version including Java 18.

**Java Technologies:**

* Java SE (Standard Edition) - Having all kinds of api to create Desktop Apps
* Java EE (Enterprise Edition) - Having multiple frameworks to create web applications
* Java ME (Micro Edition) - For mobile applications

**Difference between JDK, JVM and JRE?**

* JVM - Java Virtual Machine is responsible to convert byte code to machine code. JVM is part of JRE and JVM perform - Load Code, Convert Code to machine code and execute code. In short it provides RunTime environment (with the help of JRE) which makes Java as machine independent language. JVM is platform Independent
* JRE - Java Runtime Environments - JRE provides an environment to run the Java code on your JVM. It provides additional libraries which help JVM to convert, load and execute the Java code. JVM comes inside your JRE.
* JDK (Java Development Kit) - JDK is software development kit to develop softwares in Java. JDK more components along with JRE like Compilers, Debugger, Java Doc etc. JDK is like complete package which has JRE and JVM as well.

**Difference between JDK, JRE and JVM**

JDK - Java Development Kit - Complete software Kit which you need if you want to develop any java software. It comes with JRE and JVM by default.

JRE - Java Runtime Environment - its required if you want to install any java based application.

JVM - Java Virtual Machine - It’s a logical program which is responsible for executing your programs. Its already in built with JDK or JRE.

**How Java works?**

Java is known to write once run anywhere which means you just need JVM on your machine. JVM comes with JRE which has JVM along with some java libraries to run your code.

* The program you write in Java is saved as .java extension
* Once you compile the file, Java Compiler converts from .java to .class file (Byte code
* Then JVM starts with first file (main method) and converts byte code to Machine code (Binary code)
* Same way JVM revert back to user with the results after processing

**What is Literals**

**Topic 3: DataType in Java**

* Primitive Data Type - Pre-defined data type by Java in terms of data size and character
* Non-Primitive Data Type - Not defined by Java but by developer
  + Primitive Data Type (Numbers (int - int, byte, short, long and float - float, double), Boolean (True/False), Character(ch)
    - Byte - 8 Bit (-128 , 127)
    - Short - 16 Bit(-32768, 32767)
    - Int - 32 Bit
    - Long - 64 Bit
    - float - 32 Bit
    - double - 64 Bit
    - boolean - 1 Bit (True/ False)
    - Char - 16 Bit
* Non-Primitive Data Type (Arrays, Class, Collection, Interface)
* String is actually a class but we use a data type and keep under Primitive or non-Primitive

**Define Variables in Java?**

* Local Variable - Variables defined inside a methods, constructors or block are considered as local variable. Theses variables are initialized in methods and destroyed once method work is completed
* Instance Variable - Instance variables are defined inside class but outside a method. These variables are initialized once class is instantiated. This type of variables can be accessed by any method or constructors.

**Topic 4: Operators in Java**

* Arithmetic Operator - \* / + -
* Bitwise Operator - &, ^, |
* Logical Operator - &&, ||
* Relational Operator - >, < <=, >=, ==,

**What are Access Modifiers in Java?**

* We have 4 types of Access Modifiers in Java
  + Public - Within and outside Class and package
  + Default - Within Class and within Package
  + Protected - Within Class and Package and outside package only within sub class/child class
  + Private - Within Class only. - To access it we can use getters and setters

**Difference between For Loop and While Loop**

- For loop will run for limited amount of time (defined in the loop) whereas While loop run till the condition is met. There is no pre-defined count mentioned in while loop.

**Topic 5: What is Heap Memory?**

* There are 2 types of memory in Java - Heap Memory and Stack Memory
* Heap Memory allocates and stores memory for non-primitive data(String, Class, Object) types whereas stack is used to store for primitive data types (int, float etc)

**What is collection Framework in Java?**

* Java provides several class and interfaces to deal with different kind of data this is called Java Framework collection.
* Under Collection framework, we have:
  + Iterable Interface
  + Collection Interface - Collection implements Iterable Interface
  + List, Queue and Set interfaces comes and implements Collection Interface
  + List has 3 most important class like ArrayList, LinkedList and Vector
  + Set has HashSet, LinkedHashSet and SortedSet

**What is Class and Object in Java?**

* Java is an Object Oriented Programming which means every variable and method are completely based on an object. Taking an example of real world, things all around us are considered as an object. Now an object can have its state and actions similarly an object is Java an instance of a class which can have its properties in terms of different variables and their methods. We can create multiple object against a class. We can consider Noun is an object, Adjectives are noun’s state and verb is like their actions.

**Different ways to create an object?**

* Using new Keyword
* Using new instance() - Depricated
* Using Clone() method
* Using Deserialization

**Topic 6: What is Constructor in Java?**

* Constructor is member method of an object
* Constructor will always have same name of Class Name
* Constructor never have return type.
  + We can make contractor as private also but in that case we need to use getter and setter to access it.
* Constructor will also have access modifiers
* Constructor will be used to allocate memory
* Every time when we create an object Java calls default Constructor
* Default constructor is created by Java during compilation only for memory allocation
* We can have Constructor overloading by passing different combination of parameters
* JVM will provide a default constructor once we create any object, but when we define any parametrized constructor then JVM won’t call the default one.
* There are 2 types of Constructor:
  + Default/ Implicit Constructor - Created by Java for memory allocation
  + User Defined/ Explicit Constructor - Created by users and we can provide parameter inside the constructor.
* Constructor Chaining - We can call a constructor within another Constructor. To call constructor simply use either new constructorname(); or this();

**What is OOPs in Java?**

* Polymorphism - Method overloading and Method overriding
* Inheritance - Parent class and child class
* Encapsulation - Idea is combining data and method under 1 unit is called Encapsulation
* Abstraction - Showing important data to user and hiding unimportant data from user

**What is Inheritance in Java?**

* Inheritance is one of important concept of OOPs. Using inherit.. an object access all properties and methods of another class. Its like child can access its parent properties.
* Things to remember in Inheritance:
  + **Class** - Class in Java is a blueprint or template from which objects are created inside the class. Like Fruit is a class and Mango, Guava or Orange are its object types.
  + **Parent Class/SuperClass** - A class becomes Super Class or Parent Class once another class extends/calls it and use Superclass properties and methods.
  + **Child Class/ Sub Class** - A class becomes or gets called as sub class or child class when it extends another calls and use its property and method.
* **Types of Inheritance:**
  + Single - In this we has 1 Parent and 1 Child Class Inheritance
  + Multi-Level Inheritance - In this we have multi-level inheritance - like chain (Parent-child-grand child)
  + Hierarchical Level - When two or more classes inherits a single class, it is known as *hierarchical inheritance.* One Parent class -2 child class
  + Multiple - Child is inheriting from different parent classes - **Not Supported in Java**
  + Hybrid - Parent - various Child - sub-child
  + Child class can’t extend more than 1 parent
  + But Child class can implements more than 1 interfaces

**What is Polymorphism?**

* As we know English Literature and vocabulary is a combination many good languages. PolyPolymorphism has come from Greek work which is a combination of 2 words Poly and morphism. Poly stands for Many and morphism stands for shapes. In Java, when a method is created with same names with different parameter combinations its called and kept under one of the OOPs concept i.e. Polymorphism. We have 2 types 1 is compile time i.e. Method overloading and 2nd is run time i.e Method overwriting which is run time.
* Runtime errors are worst as it won’t deduct during compile time and it will appear during run time of the program.

**Difference between Method and Constructor**

* Constructor has always same name as Class Name - Method can have any name
* Constructor never return any value - Method may have return value;
* Constructor may accept various parameter same as Method

**7. What is Method overriding?**

* When we use/define same method name in multiple classes (Parent and child class)
* The method should have same return type and accepts similar parameters
* The body of method can be different or modified in child class method
* During Runtime, Java complier will decide based on condition that which method needs to be called that the reason we consider Method Overriding as Runtime.

**What is Encapsulation in OOPs ?**

* Encapsulation allows us to bind class variables and methods in a single unit. It also allows to hide class variables from other classes.
* Reference variable should always declared as private and methods should be declared as public.
* These private variables can be accessed by creating public method called as getter and setter and force to get the data
* Getter and setter are not pre-defined methods, its user define method like normal methods where we will start method name with get and set (getAccountNo() or setatmPin()). Get method we normally has return type whereas set will have void (no return type). The objective of getter and setter methods is mainly get or set the value of private class variable.

**What is Abstractions and Abstract in OOPs?**

* Abstraction is one of the 4 pillars in OOPs. Abstractions is a concept based on that we declare method but we don’t implement fully that time as we don’t know all the details. Later we implement those methods.
* Abstract keyword is a keyword which is used as non-access modifiers for classes and methods.
* If we use Abstract keyword before any class then we can’t create any object of that class
* If we use Abstract keyword before any method then that method must be overridden in first concrete class. (Concrete - Immediate child class)
* Abstract Method uses abstract keyword prior to method name and doesn’t have any body

**What is the difference between Abstract Class and Interface?**

* Abstract Class may have abstract methods (no method body defined) and concrete methods (where method body is defined) - basically depends on developer how many methods they want to define as abstract or how many as concrete.
* Interface will have by default all methods as abstract.
* For Abstract class, the immediate child class will extend the parent class (the abstract class) and must implement all the unimplemented methods
* Abstract methods must be implemented to its child class and its only limited to it. If Child class have another child of it then it’s not required

**How can we call a concrete method (non-abstract method) of abstract class as we can’t create an object of an Abstract class?**

* We can do by inheriting to child class and child class object can access that method.

**What is Static Keyword in Java?**

* Static is a keyword in Java which can be used for variables, methods or block. When information is common, we use static for it. Like for Institution, school name will be static for all students.
* For static variables and method, memories are defined only once but for objects memory is allocated every time once we create an new object.
* To access any static variable and and methods, we don’t need to create any object instance to access, we can call it directly at class level.
* Static method can't call non-static method but non-static method can call static method

**What is Method and Constructor Overloading**

* When Method or Constructors are created with same name but different combination of arguments, its called Constructor or Method Overloading
* Method overloading may have different combination based return type and different arguments
* But Constructor doesn’t have return type so only we can use different arguments to differentiate

- This is also called as parameterization

**What is Final keyword?**

* Final keyword can be used before any variable, method or class
* Before Variable - the assigned value can’t be changed
* Before method - the method can’t be override in inheritance class
* Before class - the class can’t be inherited by another class

**What is super Keyword in Java?**

* When we have same methods in child class and parent class and we call that method, JVM gives preference to child class first or the local class from which we are calling that method. And parent class method won’t be get executed. Using Super keyword we can call parent class method also.
* So super keyword is basically used to get the override method of super class.
* There must be a Inheritance to use Super keyword

**What is Interface?**

* Interface is kind of blue print of a class.
* Interface will have only abstract methods, we can’t have concrete method inside interface
* Interface will have default all methods as abstract so no need to mention as abstract keyword
* We can’t use final keyword before abstract class or interface as final keyword doesn’t allow to override the method whereas interface and abstract class works on the logic of override.
* Interface methods are by-default public
* Interface methods are by-default abstract
* Abstract method means - its only definition and not implementation
* We can’t instantiate Interface we can use Class to do so for example

Test\_interface obj = new Test\_Class();

- We can implements multiple interface using comma like Test\_class implements test\_interf1, interf2, interf3 etc.

**What is packages and how to create user defined packages?**

* Packages are nothing but it's a collection of classes.
* Packages are 2 types - User defined and System defined
* User defined - means we can create package inside a package like p1.p2.p2
* System defined - when we import any package like for ChromeDriver, we import its package

**What is Exception Handling and how many types of it?**

* Exception is an event which breaks the code and terminates our program unexpectedly.
* Two types of exception
  + Checked Exception - A user error which is typically can’t be foreseen by the programmer like file is not available on location etc.
  + Unchecked - RunTime Exception - While executing the code, this error can appear, this may probably be tracked and corrected by programmer during compilation. In some cases though this may be skipped in compilation can appear during runtime
  + Generally all executional/logical code we will put inside try and based on potential error type we can define Exception and capture the message.
  + Unchecked -Runtime Exceptions are - Arithmetic Error(divide by 0), Array (out of bound), Security, Index out of bound, File format
  + Java Checked Exceptions - Class Not found, No such element found, File not found, Illegal Access
  + Exception Hierarchy : Throwable -> Error, Exception -> Exception -> IO Exception, Runtime
  + We can put the code inside try block and use catch block to print the exception printout.
  + Note, if no exception is found and code is clean, no exception message will be printed out.
  + We case multiple try and catch blocks for different codes
  + Syntax for try and catch:

try

{

all your code

}

catch(ArithmeticException e)

{

System.out.println(“Your message”);

}

catch(otherexceptions e)

{

System.out.println(“Your message”);

}

catch(Exception e)

{

System.out.println(“Your message”);

}

finally

{

System.out.println(“Your message”); - This code will always execute no matter what condition is.

}

**How to create and read .properties file in Java and Selenium?**

* In Selenium we keep all common information like urls, userid, password, xpath of all the properties of a page which we want to automate in property file. Then in code use the key (name = “Santosh” here name is key) to get this data.
* Create a new file type under any folder and name with extension as .properties and then save all the key and its values inside it
* Create a Java Class or in Selenium your test case class file and to read this properties file, we need to use Properties class and create an object of this.
* Create FileInputStream Class and its object to and pass the path of this properties file inside it
* Then simply use Properties predefine load method and provide FileInputstream object as parameter inside it.
* To read the data simply use Properties class object and use method as getProperty(“key”);

**How to read and write a file in Java or Selenium - Stream Channel?**

* Stream is kind of a channel which is used to read and write a file.
* FileInputStream is read the data and FileOutputStream to write a file

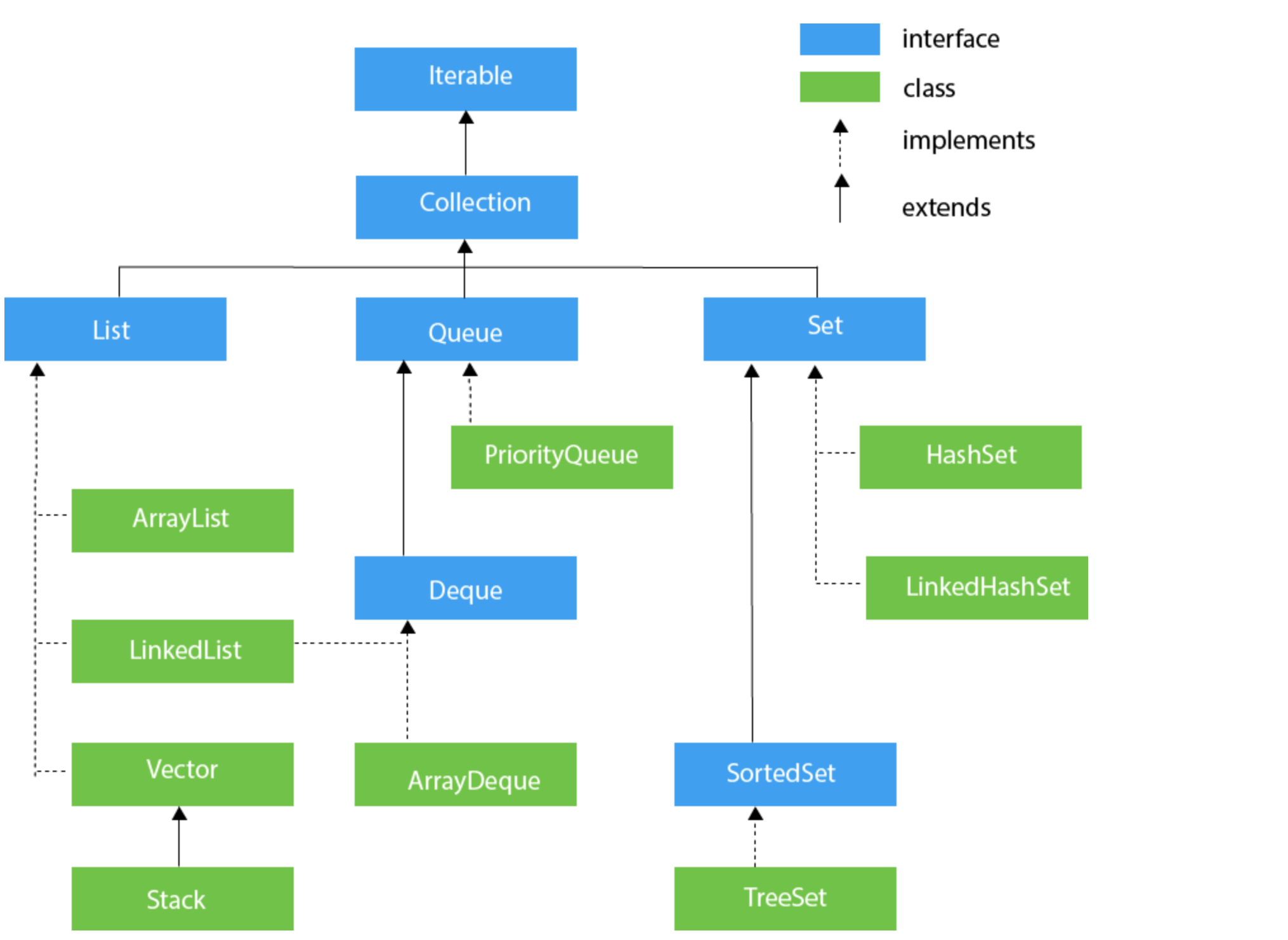
**How to read and write a Text File and CSV File in Java?**

* We need File Class to locate the location and provide the file name
* For Text and CSV both we follow almost same approach, use right extension .txt or .csv
* We FileWriter/Read Class and BufferedWriter/ BufferedRead class to write and read the files

**How to write an HTML File in Java?**

* We need File Class to locate the location and provide the file name
* For HTML we follow almost same approach like text file use right extension .html
* Use FileInputStream class to handle the file
* We FileWriter/Read Class and BufferedWriter/ BufferedRead class to write and read the files
* Using write() method use HTML tags and attributes in HTML format (the way we format to create a html page)

**Java Collection Framework Diagram?**



**Java Collection Framework?**

* Java Collection framework defines different data structure and how to store and perform action on these.
* Collection is a high level interface which has multiple sub interface available like Set, List, Queue, Dequeue. Map comes from different hierarchy as it stores different set of data.
* For Set, we have Hash-Set mainly used and Tree-Set are commonly used implementation
* For List, we have Array List mainly used and LinkedList which common used implementation
* List-list implementation can used be used for both Array List as well as DeQueue
* For Map, we have Hash map
* For Queue, LinkedList is mainly used
* For Dequeue- ArrayDequeue is mainly used

**List - List Classes?**

**A. ArrayList** - ArrayList is implements List Interface and implemented inside Java.util.list interface

* + - * Syntax : ArrayList<String/Integer>names = new ArrayList<String/Integer>();
      * ArrayList Methods : add(), remove(), set(), addall(), clear, contains, index etc- to add one array to another
      * ArrayList is backed by Arrays
      * ArrayList size grows dynamically as we add more data means JVM provides bigger size in memory
    - ArrayList comes with plenty of inbuilt methods like add, remove, copy etc.
    - Array comes with 2D and 3D whereas ArrayList not and Array is faster compare to ArrayList

**B. LinkedList -** LinkedList implements the List interface. It uses a doubly linked list internally to store the elements. It can store the duplicate elements. It maintains the insertion order and is not synchronized. In LinkedList, the manipulation is fast because no shifting is required.

**- Syntax:** LinkedList<Integer>num = new LinkedList<>();

**- Methods :** add, pop,addFirst, addLast, clear, clone, contains

**C. Vector -** Vector is a Class which implements List Collection Framework.

- **Syntax** : Vector<String> name = new Vector<>();

- **Methods** : add, pop,addFirst, addLast, clear, clone, contains

**D. Stack -** Stack is a class under List Interface. In Stack the number order work as Last in - First out for example keeps books on stack.

* + - * **Syntax** : Stack<String> name = new Stack<>();
      * **Stack Methods** : push(), pop(), peek(), clear();
      * We use push and pop to add and remove element. Push to add and pop to remove.
      * We have few other Stack method which we can use and play with Stack data

**16. Queue - Class & Interface**

**A. Queue -** Queue works like realtime queue process i.e. first in first out.

* + - Syntax : Queue<String> name = new LinkedList<>();
    - Queue Methods : offer() - to add, poll() - remove, peek() - to look next one out
    - Queue is class under Queue Interface
    - We can use offer method to add element and poll method to remove element from queue
    - We can also use add and remove methods for same jobs but add and remove will give exception if element not available.
    - Queue uses LinkedList data type as example:

**B. Priority Queue-** PriorityQueue allows us to set the priority of the element. If we have Integer datatype then it will store the values lowest first and largest last.

- Syntax : PriorityQueque<Integer> rollno = new PriorityQueue<>();

- Methods : offer(), poll(), add(), remove()

- Add and remove method will give exception if the job is not executed properly.

- Offer and poll methods won’t give exception it will give you true or false

- We can use Comparator.reverseOrder to reverse the value

**C. Deque (Interface) -**

**D. ArrayDeque -** ArrayDeque allows you remove data from front and back both.

- Syntax : ArrayDeque<Integer> num = new ArrayDeque<>();

- Method : offer, add, remove, addFirst, addLast, removeFirst, removeLast, poll

**16. Set - Class & Interface**

1. **HashSet**

**- Syntax :** Set<Integer>num = new HashSet<>();

- Methods: add, remove,

- Under set, order is not defined so it will display elements in any order.

- In HashSet, internally java assign some key value for each element and it remains same for that element and doesn’t allow to store duplicate value.

1. **LinkedHashSet**

**Syntax :** Set<Integer>num = new HashSet<>();

- Methods: add, remove,

- Under set, order is defined so it will display elements in any order.

- In HashSet, internally java assign some key value for each element and it remains same for that element and doesn’t allow to store duplicate value.

**17. HashMap in Collection Framework**

* HashMap allows to store key and value. The way you define your variable type, same way you have store the data. The difference between HashMap and HashMapTree is sorting. HashMapTree provides you sorting.
* Syntax : Map<Integer, String> MapDemo = new HashMap<>();
* Methods : put(), remove(), clear(), containsValue(), IsEmpty()
* To read data from HashMap is below syntax:
  + for(Map.Entry<Integer, String>e:MapDemo.get.entryset())

{

print(e.getkey());

}

**18. What is Arrays Class and Framework Class?**

* Arrays and Framework class are also part of Collection framework and provide additional method to work with.
  + Syntax : int[] num = {1,2,3,4};

Arrays.sort(num);

* Arrays Class Methods: Arrays.sort(), binarySearch(), asList(), compare() etc.
* Array Class works on Array (not Arraylist)

**What is Reflection Api in Java?**

* Reflection api is used to get Class and its methods and other attributes details. Using Reflection api, we can fetch information about the class like what is the class name, what are different methods we have inside that class etc.
* We can use mutilple pre-defined methods to get these information and use the same in TestNG framework to work with it.