

Ques1. What is JAVA ?

Ans. Java is a highly level object Oriented, robust, Secure platform independent, highly performance and portable programming language it was developed by James Gosling in 1991. it is also known as platform as it provides its own JRE and API.

2. Features of Java :-

1. Simple :-

Java is easy to learn the Syntax of java is base on C++ which makes it easier to write program in it.

2. Object Oriented :-

Java follows Object-Oriented paradigm which allows us to maintain our code as the combination of different types of objects.

3 Portable :-

Java is write once and run anywhere we can execute java program on every machine java program (.java) Converted to bytecode (.class) which can easily run on every machine.

4. Platform :-

independent Java comes with its platform on which its code is executed. Java doesn't depend upon the operating system to be executed.

5 Secured :-

Java is secured because it doesn't use explicit pointers. Java also provides concept of byte code and exception handling which makes it more secured.

b. Robust :-

Java is strong programming language as it uses strong memory management concept like automatic garbage collection. Exception handling make it more robust.

7. Architecture Neutral :-

Java is architecture neutral as it is not dependent on architecture in C1 size of data types may vary according to architecture (32 bit or 64 bit).

g. Interpreted :-

Java uses JIT interpreter along with the compiler for program execution.

9. High performance :-

Java is faster than other traditional languages because Java bytecode is close to native code.

10. Multi-threaded :-

The main advantage of multi threading is that it doesn't occupy memory for each thread it shares common memory area.

11. Distributed :-

RMI and EJB are used for creating distributed applications.

12. Dynamic :-

Java is dynamic language it supports dynamic loading of classes it means classes are loaded on demand.

Ques 3 What do you understand by JVM?

Ans. JVM is a virtual machine that enables the computer to run Java program. JVM acts like a run-time engine which calls the main method present in Java code. The Java code is compiled by JVM to be a bytecode which is close to native code.

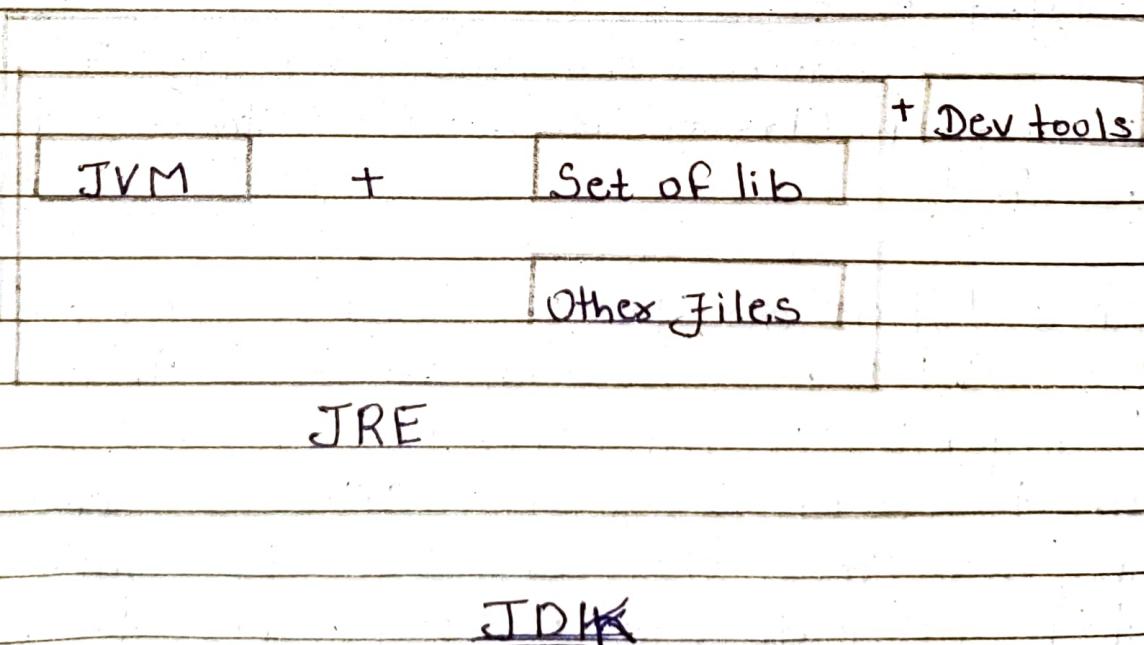
Ques 4 How many types of memory allocated by JVM?

- Ans i> Heap - Whenever object is created.
ii> Class Area
iii> Static
iv> Native method Stack
v> Program Counter register.

Ques 5 What is platform?

Ans Platform is a set of hardware and software environment in which piece of SW is executed.
Java is software based platform.

Ques JDK | JRE | JVM



JDK

JRE

JVM

i) Java development kit	Java Runtime environment	Java virtual machine.
ii) Platform dependent	Platform dependent	Highly platform dependent.
iii) It is superset of JRE	It is a subset of JDK	It is a subset of JRE.
IV) JDK Comes with installer	JRE only Contains environment to execute Source Code.	JVM bundled in both JDK and JRE.
V) It Contains dev tools, debug tools and other tools	It Contains set of libraries and other supporting JAR to execute Java Code.	It does not contain any tools.

Ques JIT Compiler?

At Compile time.

Source Code : Java → Compiler → Byte Code

At Run-time.

Native machine code ← JIT Compiler

The Java runtime environment (JRE) is responsible for executing source code and it contains JIT Compiler that does the performance optimization. It converts the byte code into native machine code at runtime.

Ques 8. What gives Java its write once run anywhere nature? (WORA)

Ans The bytecode - Java compiler converts Java source code file (.class) into byte code which is intermediate language between source code and machine code. The bytecode is not platform specific and can be executed on any computer.

Ques 9. Is Empty .java file name is valid?

Ans Yes

Class A {

 - - -

}

To Compile :- javac .java

To run :- java A

Ques 10. What is the default value of local variables?

Ans. Local variables are not initialized to any value, not primitive nor object references.

Ques 11. What are the different access specifiers in Java?

Ans Public, Private, Protected, default

- Public → Within Class, package, outside package, by subclass.
- Private → Within Class Only
- Protected → Within Class, package, outside package by subclass only.
- default → Within Class & package.

Ques 12. What is the purpose of static methods and variables in Java?

Ans. The methods or variables that are defined as static are shared among all the objects of the class. the static is a part of class and not the objects. the static variables are defined in class area and we don't need to create object of class to access such variables.

Ex:- College name of Student

Ques 13. What are the advantages of packages in Java?

Ans

- i) Packages avoid name clashes
- ii) packages provide easier access control.
- iii) We can have classes access only within the package.

Ques 14. What is the initial value of object references which is defined as an instance variable?

Ans.

All Object references are initialized to null.

Ques 15. What is Constructor?

Ans. Constructor is a special type of method that has same name as of class name and is used to initialize state of object. every time object is created using new keyword, default constructor is called it must not return any explicit value.

Ques 16. How many types of Constructor are used in Java?

Ans.

Default Constructors, Parameterized Constructors.

Ques 17. What is the purpose of default Constructor.

Ans.

```
Class Student {
    int id;
    String name;
    Student_info()
    {
        S.O.P (id + " " + name);
    }
}
```

```
P.S.V main (String args[])
{
```

```
    Student s1 = new Student();
```

```
    Student s2 = new Student();
```

S₁. Student.info();

S₂. Student.info();

}

}

Output :- O null

O null.

Ques 18. Does Constructors return any value?

Ans

Instance of Class.

Ques 19. Can Constructors inherited?

Ans No, Constructors cannot be inherited in java

In inheritance SubClass inherits the members of a Super Class except Constructors.

Ques 20 Can you make Constructor final?

Ans No, A Constructor can't be made final. A final method can not be overridden by any subclasses. As mentioned previously, the final modifier prevents a method from being modified in a subclass. The main intention of making a method final would be that the content of the method should not be changed any outsider.

Ques 21. Can Constructor overloaded?

Ans Yes, by changing no of arguments or by changing data type of arguments.

Class Test {

int i;

Public Test (int k) {

i = k;

}

Public Test (int k, int m) {

i = k + 1;

m = k + 2;

}

P.S.V.main (String [] args) {

Test t₁ = new Test (10);

Test t₂ = new Test (10, 20);

S.O.P (t₁);

S.O.P (t₂);

}

}

Ques 22. Static variable and static method

Ans Static variable used to refer to common property of all objects.

- Static method belongs to class rather than object.
- It can be called using class name.
- A static method can access and change value of static variable.

Ques 23. Write difference between Constructor and method.

S.No	Constructor	Method
1.	Constructor must have same name as of class or name.	Method name may or may not be same.
2.	It must not have any explicit return type.	It should/must have a return type.
3.	Constructor can not be overridden.	Method can be overridden.
4.	Constructor is used to initialize state of object.	Method exposes the behavior of the object.
5.	Constructor is invoked automatically when the object is created using new keyword.	Method needs to be explicitly invoked.
6.	Java Compiler provides default constructor if you don't have any constructor in class.	Compiler does not provide any default method.

Ques 24. What are the restrictions applied to Java Static Ans methods?

Ans

- i) Static method can not use non static variable or call non-static method directly.
- ii) this and Super keyword cannot be used in static context as they are non static.

Ques 25. Why the main method is static?

Ans

Because objects are not required to call the static method. If we make java main method non-static, JVM will have to create object first and then call main() method which will lead to extra memory allocation.

Ques 26. Can we override static method?

Ans No, we cannot override static methods because method overriding is based on dynamic binding at runtime and the static methods are bonded using static binding at compile time.

Ques 27. Can we execute Java program without main() method?

Ans It is possible before JDK 1.7. Since JDK 1.7 it is not possible.

Ques 28. Can we overload static method?

Ans. Yes

Class Test {

 static void display (int a) {
 S.O.P (a);

 Class Test {

 Static void display (int a, int b) {
 S.O.P (a + " " + b);

 P.S.V.main (String args[]) {

 Test.display (10);

 Test.display (10, 20);

Ques 29. Can we make Constructors Static?

Ans

Constructor is invoked when object is created if we try to make Constructor static, it will throw compile-time error

Ques 30. Can we declare static variable inside

Ques 31. Can we make abstract methods static?

Ans In java, if we declare abstract methods as static then it will become part of class and we can directly call it which is unnecessary calling undefined method completely useless therefore it is not allowed.

Ques 32. What is this keyword in java?

Ans 'this' keyword is a reference variable that refers to current object.

- i) this can be used to refer current class instance variable
- ii) this can be used to invoke current class method.
- iii) this can be used to invoke current class constructor
- iv) this can be used to return current class instance from the method.
- v) It can be used as argument in method call.
- vi) It can be passed as argument in constructor call.

Ques 33. Can 'this' is used to refer static members?

Ans. Yes, but it is not good programming practice

Ques 34. Which class is super class of all classes? →

Ans. Object Class

Ques 35. Why pointers are not used in Java?

Ans. Because pointers are unsafe and complex to understand.

Ques 6. What is inheritance?

Ans. Inheritance is the mechanism by which one object acquires all the properties and behaviour of another object of another class it is used for code reusability and method overriding.

- i) Single level
- ii) Multi-level
- iii) Multiple
- iv) Hybrid
- v) Hierarchical

Advantages:-

- i) Inheritance provides code reusability.
- ii) To achieve runtime polymorphism.
- iii) Inheritance provides data hiding. the base class can hide some data from the derived class by making it private.
- iv) Code is less and more readable.

→ Super can be used to refer to immediate parent class instance variable.

→ Super can be used to invoke immediate parent class method.

→ Super can be used to invoke immediate parent class constructor.

Ques 37. Can we use this and super in a Constructor?

Ans. No, this must be first statement in a Constructor.

Ques 38. Can we override the overloaded method?

Ans. Yes

Ques 39. Can we override private methods?

Ans. No, because scope of private method is limited to the Class and we cannot access them outside the Class.

Ques 40. Can we change the scope of overridden method in Subclass?

Ans Yes, we can change the scope of overridden method in the Subclass but we cannot decrease the visibility.

- Private :- Private can be changed to public, protected and default.
- Default : default can be changed to public.
- Protected :- Can be changed to public and default.
- Public :- Will remain public.

Ques 41. Final Variable / Class / method.

- Ans
- Stop value change.
 - Stop method overriding.
 - Stop inheritance.

Ques 42. Can we modify 'throws' clauses of superclass method while overriding it in subclass?

Ans Yes but -

- i) If Superclass doesn't declare exception, Subclass overridden method can not declare checked exception but can only declare unchecked exception.
- ii) If Superclass declare exception, then subclass cannot declare parent exception of exception declared by Superclass.

Ques 43 Final blank variable.

Ans A variable declared as final and is not initialized

Ques 44. Can you declare main method as final.

Ans. Yes, P.S. final void main (String args[]).

Ques 45 Can we override main method.

Ans. No, because main method is also a static method.

Ques 46. Can we declare Constructor as final?

Ans. The Constructor can never be declared as final, because it is never inherited if you try to do so, it will throw Compile time error.

Ques 47. Can we declare interface as final?

Ans. No, you cannot declare interface as final because interface must be implemented by some class to provide its definition therefore, there is no sense to make it final however if you try to do so it will throw compile time error.

Ques 48. Can we declare final method in abstract method

Ans. Abstract method can not be final as we need to override them in subclass to give its definition.

Ques 49. What is java instance of operator?

Ans. The instanceof in Java also known as type Comparison operator because it compare instance with type.

Class Simple {

P.S.V.main(String args[]) {

 Simple s = new Simple();

 S.O.P(s instanceof Simple);

}

}

Ques 50. Can we achieve runtime Polymorphism by data members?

Ans. No, we can override data functions, not data members.

Ques 51. What is abstraction?

Ans. Abstraction is the process of hiding the implementation details and showing only functionality to the user.

Abstraction enable you to focus on what object does instead of how it does.

- Interface
- Abstract Class.

Ques 52. What is the difference betn abstraction & encapsulation?

Ans Abstraction hides the implementation details encapsulation wraps the code and data into single unit.

Ques 53. What is abstract Class?

Ans. A class that is declared as abstract is known as abstract class. it needs to be extended and its method needs to be implemented it can not be instantiated.

It can have abstract, non-abstract static methods and Constructors.

It can also have final methods which will force subclass not to change body of method.

Ques 54. Can there be an abstract method without an abstract class?

Ans No, if there is a abstract method in a class, the class must be abstract.

Ques 55. Can you use abstract and final both with a method?

Ans No, because we need to override the abstract method to provide its implementation whereas we can't override the final.

Ques 56. Can you declare interface method static?

Ans No, because methods of interface are abstract by default and we can not use static & abstract together.

Ques 57. Can interface be final?

Ans No, because interface needs to be implemented by other class and if it is final, it can't be implemented by any class.

Ques 58. How many types of exception can occur in program?

Ans i) Checked - Exception - Compile time, SQL exception, ClassNotFoundException.
ii) Unchecked exception - Runtime, ArrayIndexOutOfBoundsException, NullPointerException.
iii) Error :- Errors cause the program to exit since they are not recoverable - OutOfMemoryError, StackOverflowError.

Ques 59. Hierarchy of Java Exception classes.

Throwable

Exception

Error

IOException

StackOverflowError

SQLException

Virtual machine Error

ClassNotFoundException

Out of Memory Error

Arithmatic Exception

ArrayIndexOut of Bounds

String Index Out of Bounds

Null pointer Exception

NumberFormat Exception

• Unchecked Exception:-

The Classes that extends runtime exception are known as unchecked exception, they are not checked at compile time.

• Checked Exception:-

The Classes that extends throwable classes except RuntimeException and error are known as checked exception.

Ques 60. What are base class of error and exception?
Ans. Throwable.

Ques 61. Is it necessary that each try block must be followed by catch block?

Ans. It should be either follow by catch block or by finally block.

Ques 62. What is finally block?

Ans. The finally block is used to execute important code of program. it executed whether an exception is handled or not.

→ Finally block is mainly used to place the clean-up code such as closing file or closing database connection.

→ For any try block, there can be zero catch block, multiple catch blocks, but only one finally block.

→ Finally block will not be executed if program exit by calling System.exit(), by causing fatal error

Ques 63. What is String Pool?

Ans. String pool is the space reserved in heap memory that can be used to store strings. the main advantage of using string pool is whenever we create string literal, JVM checks the string pool constant first, if string already exist it return the reference of that string if string doesn't exist, it return new string reference it saves memory, avoid duplicacy.

Ques 64. What is the meaning of immutable regarding String?

Ans The simple meaning of immutable - as unmodifiable or unchangeable. In Java String is immutable, once String Object is created, its value cannot be changed.

Suppose there are five reference variables all refers to the one object, if one reference variable change value of object, it will affect all ref variable that is why String objects are immutable in java.

Ques 65. How many ways we can create String object

Ans i) String literal

String s1 = "Welcome";

ii) new keyword

String s2 = new String ("Welcome");

Ques 66. How many Objects Created?

Ans. String s1 = "India";

String s2 = "India";

→ Only one object is created. in String literal pool.

Ques 67. How many objects are created?

Ans String s1 = new String ("India");

→ Two - One in String Constant pool and other in non-pool (heap).

Ques 68. How can we create immutable class in Java?

Ans By making Class and all of its member as final.

Ques 69. Why CharArray is preferred over String to store password?

Ans String stays in the String Pool until the garbage is collected. If we store the password into the string it stays in memory over long period of time and anyone having access to memory dump can extract the password as clear text.

On the other hand, using CharArray allows us to set it to blank whenever we are done with the password. It avoids the security threat and enables us to control memory.

Ques 70. What is garbage collection?

Ans Garbage Collection is the process of reclaiming the unused runtime objects. It is performed for memory management. In other words, we can say that it is a process of removing unused objects from the memory to free up space and make this space available for JVM.

Ques 71. What is gc() method?

Ans The gc() method is used to invoke garbage collector this method is found in System & runtime classes.

System.gc();

Ques 72. How object is unreferenced?

Ans 1. By nulling
 $S_1 = \text{null};$

2. By assigning reference to another
 $S_1 = S_2;$

3. By anonymous object.
`new A();`

Ques 73. Is Java is pure Object oriented?

Ans Java uses Primitive data type and hence is not pure Object oriented language.

Ques 74. Can Class be declared as protected?

Ans. No, only methods can be declared as protected.

Ques 75. Can a source file contain more than one class declaration?

Ans Yes, source file contain any no of class declaration but only one class can be declared as public

Ques 76. Why do we need constructor when we have methods?

Ans We use Constructors to initialize the object with the default or initial state. the default values for primitives may not be what are you looking for. Another reason to use Constructor is that it informs about dependencies.

Ques 77. What is implicit return type of Constructor?

Ans. Constructor goes, it is used to initialize and object of the class. So implicitly, they return the current instance of the class. & whose constructor it is ~~is implicit~~ Therefore, implicit return type of Class Constructor is "Class type itself".

→ It returns the instance of the class to which it belongs.

Ques 78 Why only Constructor is responsible for creating an object.

Ans A Constructor is responsible for creating an object because it ensures proper initialization of the object's attributes and sets up its initial state. It centralizes the essential setup logic, promoting consistency and reducing the chances of an object being in an inconsistent or undefined state. This helps maintain the integrity and reliability of the object, making it a critical component of object creation in object-oriented programming.

Ques 79. Why Constructor Cannot be static?

Ans The Constructor in Java cannot be static because if the Constructor are marked as static, they can not be called from the child class.

→ The child class object will not be created.

→ The program will not be compile and throw a compile time error.

Ques 80. Why Constructor is called Constructor not initializer?

Ans. A Constructor in programming not only sets initial values but also prepares the entire object for use, including any necessary setup tasks. So, Constructor is used because it constructs the whole object not just initializes it like an initializer might imply.

Ques 81. When will object is created during runtime or during Class Creation?

Ans An object is created during runtime when you use the blueprint of a class. So, Object are made when your program runs not when you design the class.

Ques 82. When will default Constructor is created during runtime or during Class time.

Ans Default Constructor is created during Compile time.

Ques 83. What is String buffer and why its used?

Ans. A String buffer is like a string, but ^{in can} modified at any point in time it contains some particular sequence of characters, but the length and content of the sequence can be changed through certain method calls. String buffer are safe for use by multiple threads.

- String buffer is mutable.
- String buffer is a temporary storage area.
- String buffer is a thread safe to use multithreaded environment.

Ques 84 What is thread Safety? Why it is used?

Ans. Thread Safety is the avoidance of data races -- situations in which data are set to either correct or incorrect values, depending upon the order in which multiple threads access and modify the data.

Threads allows a program to operate more efficiently by doing multiple things at same time. threads can be used to perform complicated tasks in the background without interrupting the main program.

Ques 85 What is String builder? Why it's used?

Ans. String Builder Class can be used when you want to modify a string without creating new object. For example, using String Builder class can boost performance when concatenating many strings together in loop.

StringBuilder is mutable and when you concatenate several strings these are created in a "heap" and not in the "pool" of string saving the JVM memory and reducing garbage collection time, for that reason it's a good way to use and manipulate strings.

Ques 86 Difference b/w StringTokenizer and StringBuilder?

Ans

String Buffer

1. Objects are mutable/ can change.
2. Have append method.
3. equals() method meant for reference / address comparison.
4. default Capacity is 16.

String Builder

1. Objects are mutable/ can change.
2. Have append method.
3. equals() method meant for reference / address comparison.
4. default Capacity is 16.
5. No method present in String builder is synchronized.

Ques 87 Why and what is the use of intern method?

Ans. The intern method creates an exact copy of a string constant pool. Note:- intern() method is used to store the string that are in the heap in the string constant pool if they are not already present.

When we declare string with new keyword it will create memory in heap but if we use intern() method first it will check in string literal pool if it found it will not create another memory in heap.

Ques 88. Why equals method not check Content in StringBuffer and StringBuilder?

Ans. Because StringBuffer is mutable, and its primary use is for Constructing strings. If we want to compare Content in it we have to call #toString() and compare the returned value.

Ques 89. What is by default implementation of toString() method?

Ans. The default implementation of toString() method returns the Concatenation of class name and hashCode value to the object separated by @.

Ques 90. On what basis hashCode is generated?

Ans. hashCode() is given in such a way that it returns the hashCode number for the object based on "the address of the object".

Ques 91. What is mutability?

Ans. Mutable means changing over time or that can be changed in a mutable string, we can change the value of string and JVM doesn't create new object in a mutable string we can change the value of the string in the same object.

Ques 92. Where is string literal pool exist?

Ans. String Pool in Java is a special storage space in Java heap memory

Ques 93. Q) What is object?

Ans. An entity that has state behaviour and object is known as an object e.g. Chair, bike, marker pen, table, car, etc. it can be physical or logical (tangible or intangible).

→ An object has three characteristics:

- State:- represents the data (value) of an object.
- Behavior:- represents the behavior (functionality) of an object such as deposit, withdraw, etc.
- Identity:- An object is typically implemented via a unique ID. the value of the ID is not visible to the external user. however, it is used internally by the JVM to identify each object uniquely.

→ An object is real world entity.

→ An object is run time entity.

→ The object is an entity which has state and behavior.

→ The object is an instance of class.

Ques 94. What is Class?

Ans. A class is a group of objects which have common properties. it is a template or blueprint from which objects are created. it is logically entity it can't be physical Java contains:-

- Fields
- Methods
- Constructors
- Blocks
- Nested class and interface

Ques 93. What is static variable in java?

Ans. A static variable is shared by all objects of that class.

Ques 95. What is instance variable?

Ans. Variables declared inside a class but outside the scope of any blocks, Constructors or methods are known as instance variable in Java.

Ques 96. Why outer class can not be private, protected and static?

Ans Because static content (behavior or state) does not belong to any particular instance or object, it will be common to all objects and all objects are free to change any static field - and every change will be visible to every object.

Ques 97. Why inner class can be private, protected?

Ans. Inner Class are security mechanism in Java we know class can not be associated with the access modifier private, but if we have the class as a member of other class, then the inner class can be made private or protected.

Collection

Wrapper Classes:-

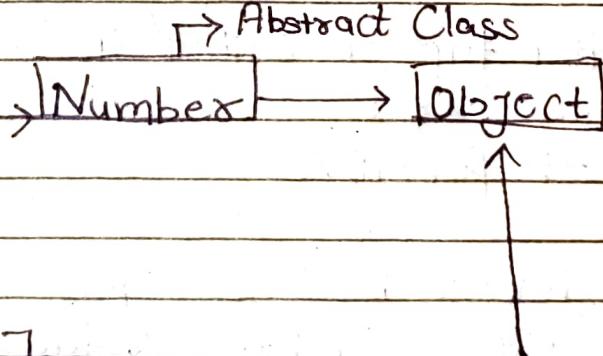
The classes which is derived from primitive datatype is called wrapper classes.

→ Just because java supports the primitive data types that's why java is called partially object oriented programming language.

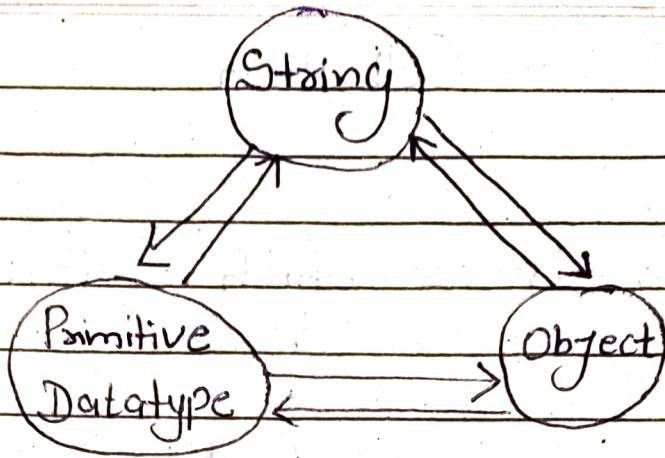
→ The language who only supports non primitive datatypes are called purely object oriented programming language.

Ex of purely OOP language is C++

byte → Byte	Abstract Class
short → Short	
int → Integer	
float → Float	
long → Long	
double → Double	
char → Character	
boolean → Boolean	



→ The process of converting primitive into its corresponding compatible object is called boxing.



1. Primitive → Object

`int x = 100;`

`Integer obj = new Integer(x);`

2. ~~glo~~ float x = 3.14f;

`Float obj = new Float(x);`

UnBoxing

→ The process of Converting Object into its Corresponding Primitive data type is called un-boxing.

Object → Primitive

1. `Integer obj = new Integer(100);`

`int x = obj.intValue();`

2. `long lobj = new Long(100);`

`long l = lobj.longValue();`

AutoBoxing.

Auto boxing is the automatic conversion that the Java compiler makes between the primitive types and their corresponding object wrapper classes.

```
int x = 100;
```

```
Integer iobj = x;
```

Auto UnBoxing.

```
Integer iobj = new Integer(400)
```

```
int n = iobj;
```

Wrapper Classes

→ You cannot convert alphanumeric String into primitive.

→ Datatype range :- It should be in the range of that primitive data type.

1. Primitive → Object (Boxing / Auto-Boxing)

2. Object → Primitive. (Unboxing / Auto Unboxing)

3. String → Primitive

```
String x = "123";
```

```
int ei = Integer.parseInt(x);
```

→ If you try to convert alpha numeric value it will generate error.

String s = 3.14 ← (xuns)

String s = 3.14F ← only & only F | f is allowed

String s = 37L ← generate an error.

→ L or any other alphabet is not allowed.

1st Way Primitive to String:-

toString() method is used to convert any datatype into String.

int x = 500;

Integer obj = x;

String s = obj.toString();
S.O.P(s);

2nd Way

int x = 500;

String s = Integer.toString(x);
S.O.P(s);

3rd Way

int x = 500;

String s = "" + x;
S.O.P(s);

`Char xc = "A";`

`Character cobj = xc;` Auto boxing.

`Char y = cobj` Auto unboxing

- Instance always set with its default value Integer
- 0. float → 0.0.

Collection

Why
What
How

Group of individual object is called Collection

→ If you have large amount of data you can store in two ways.

Array

1. Array is fixed in size

2. Memory utilization is not good.

3. Performance wise is good as compared to collection.

4. It supports primitive object.

Collection

1. Collection is growable in nature.

2. Memory utilization is good

3. It's not have good performance.

4. It supports object.

b. Homogeneous Data
(Not 100% true)

b. Collection is Heterogeneous
as well as homogeneous. (Not 100% true)

- Behind the ArrayList screen it uses the concept of array.
- Collection will create a copy.
- If you want to store different type of data in Array make use of object.

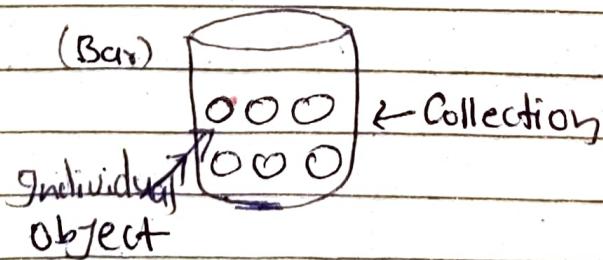
Object arr = new Object [100];

Ques What is Collection in java?

Ans Whenever we have to represent a group of individual Object as a single unit go for the Collection Concept
(O.S.)

Collection is Container object.

Real life example :- Bag, Classroom.

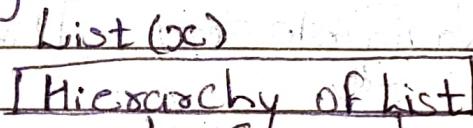


→ There are following methods of Collections :-

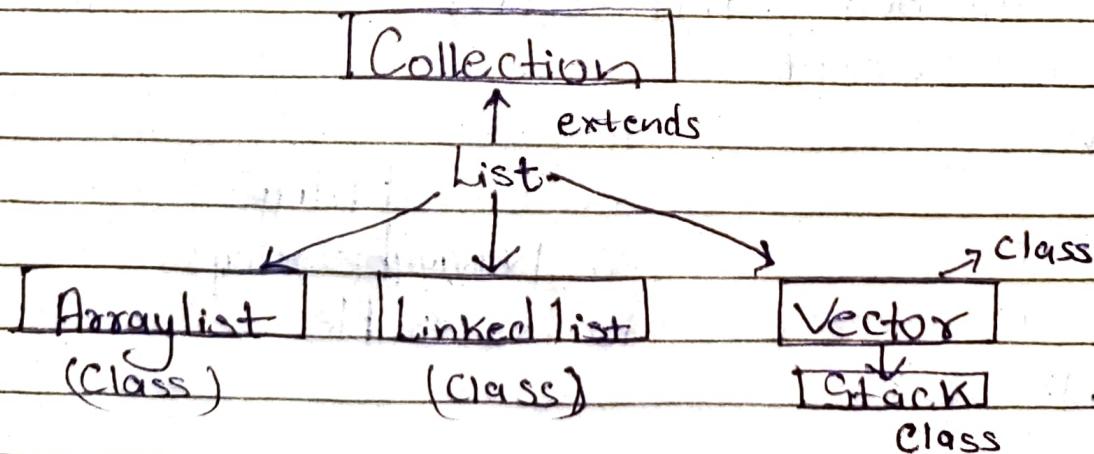
- isEmpty();
- size();
- remove();
- add();
- Contains();
- retainAll();
- addAll();
- clear();

→ The above methods are taken inside the interface that is Collection.

→ Collection is interface.



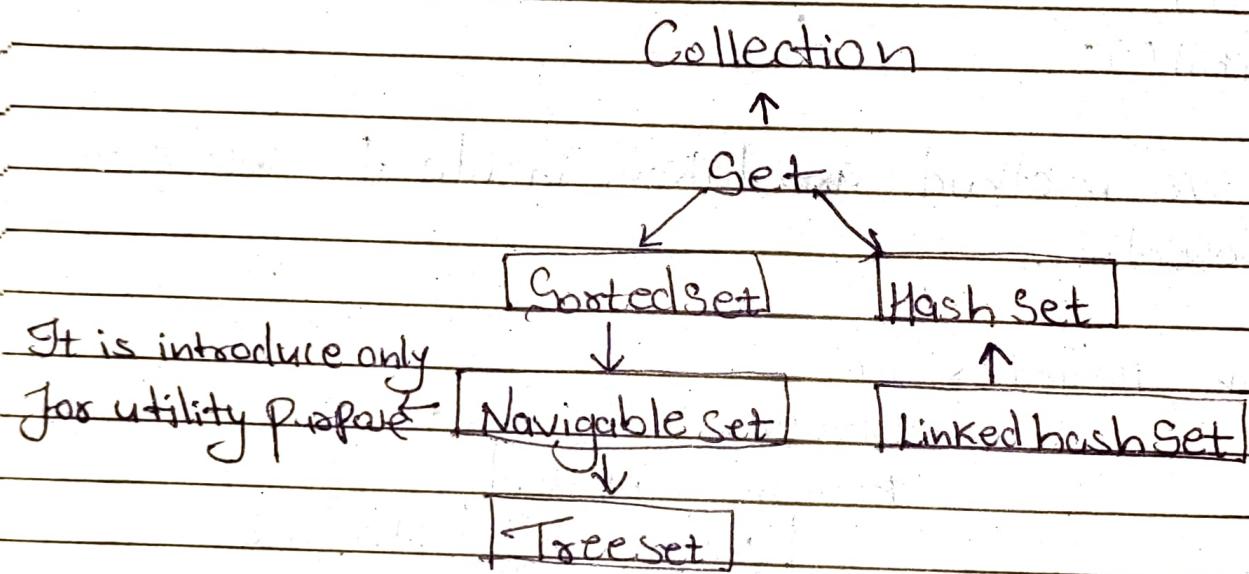
List :- Whenever we have to represent - - - single unit where duplicate data elements are allowed from list



→ Interface is backbone of Java.

Get:-

Whenever we have to represent single element where duplicate data elements are not allowed it manage data uniqueness.



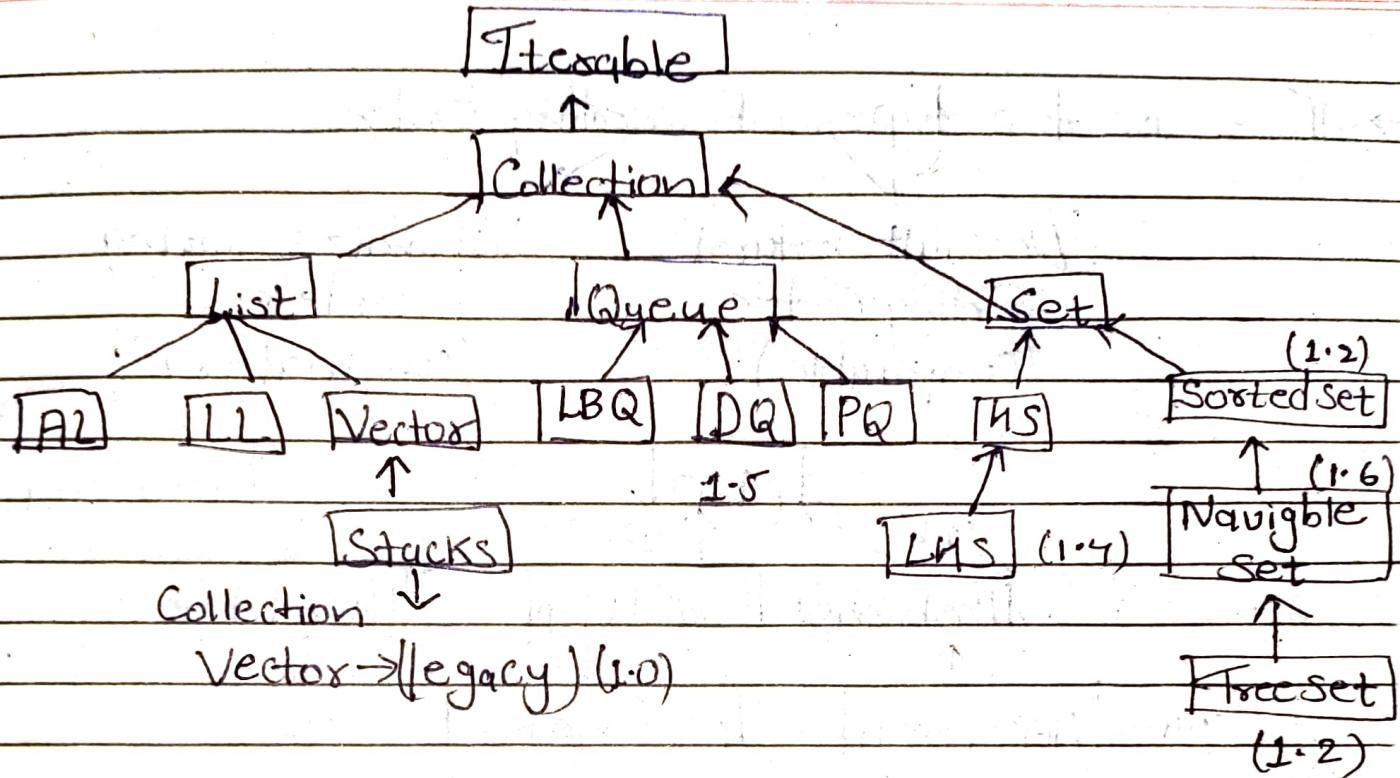
It is introduce only

for utility purpose

Sorted Set:-

Whenever we have to represent a group of objects as a single unit where duplicate data elements are not allowed and sorted set is matter go with Sorted Set Concept.

1. Collection
2. List (I)
3. Set → I
4. Sorted
5. Navigable (I)
6. Queue
7. Map
8. Sorted Map
9. Navigable Map
10. Iterable



Map:-

Whenever we have to represent `<Key, Value>` pair object as single unit go for map. Key is always unique of object. Value may be duplicate.

According to oracle :-

Map is not a true collection.

D/B Collection and Collections :-

→ Collection is interface.

→ Collections is a Class.

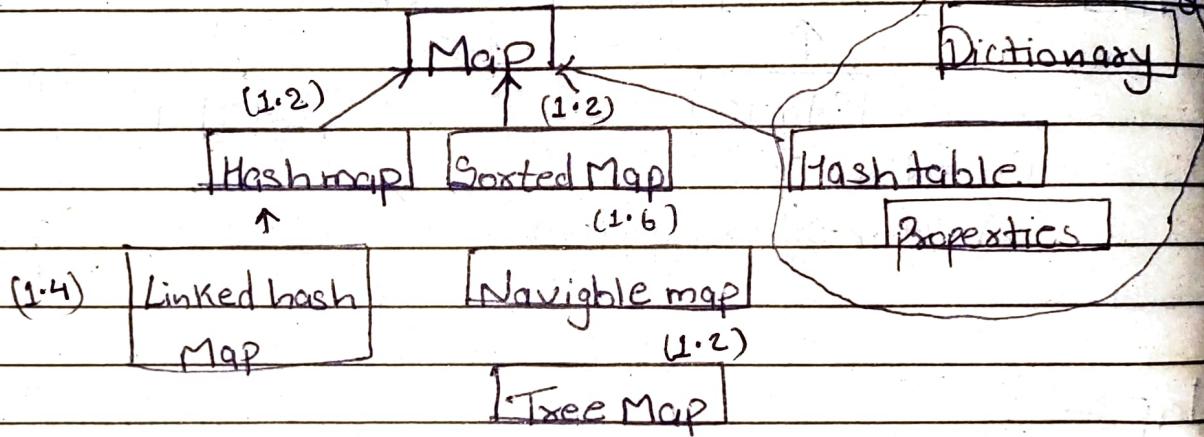
→ Set internally uses Map.

→ There are two types of Sorting Order

Default natural
Sorting Order
(Comparable)

Customize Sorting
Order
(Comparator)

I.O legacy



→ There are two types of sorting order

Default natural
Sorting order
(Comparable)

Customized Sorting
order
(Comparator)

→ At the time of object class designing the logic of sorting written by developer is called default natural sorting order Integer.

→ When a programmer write a logic to fulfill its own requirement is called Customized sorting order.

→ All the classes related to ArrayList are present inside
java.util.ArrayList

```
import java.util.Scanner;
import java.util.ArrayList;
```

```
public class P1 {
```

```
    public static void main(String args[]) {
        ArrayList<Integer> Al = new ArrayList<Integer>();
        Al.add(6);
        Al.add(7);
```

Collection
↑
List
↑
ArrayList

Java.util.ArrayList
↓
ArrayList();
ArrayList(int);
ArrayList(Collection);
↑
Sub child of Object

→ There way to declare an ArrayList.

```
ArrayList al = new ArrayList();
```

```
ArrayList al = new ArrayList(100);
```

```
ArrayList al = new ArrayList(c);
```

⇒ Abstract & interface. You can pass Child Class object
⇒ Default capacity of arraylist is 10.

- With type Safety. (object)
- Without type Safety. (Integer --- | --- | ---)

Generic Class --

A generic class is a class which can work on any type of data type or in other words we can say it is a data type independent.

→ In Java each and every class is generic by default.

```
Class Addition <T extends Number> {
    private T a;
    private T b;
    public Addition (T a, T b) {
        this.a = a;
        this.b = b;
    }
}
```

```
public void add () {
    System.out.println("Addition " + (a.intValue() + b.intValue()));
}
```

Class Test {

```
P.S.V.main (String args[]) {
    Addition < Integer > obj = new ArrayList < Integer > ();
    obj.add();
}
```

→ ArrayList preserves insertion order.

When you get the data in the same way give them it means it preserves insertion order.

→ ArrayList grows with 50%.

$$\text{cl. add}(10); \text{ ie } (C \times 3)/2 + 1 = 50\%$$

→ DB size and default capacity

ArrayList < Integer> q = new ArrayList < Integer>(100);
q.add(1); \downarrow → size ②
q.add(2); \downarrow element available.
Index must be less than equal
to size

→ These are three types bound exception in java

1. IndexOutOfBoundsException (Parent)
2. ArrayIndexOutOfBoundsException (Child)
3. StringIndexOutOfBoundsException (Child).

public C remove (int); \leftarrow (index)
public boolean remove (java.lang.Object);

1) Data value match if available it removes return
true) if it returns false)

Perfect Matching
Matching → Suitable Matching.

return type Boolean

```
import java.util.Collections;  
Collections.sort(list);
```

Cursor is a sub class of object class it can be define as cursor indicates on the screen.

→ A Java Cursor is an iterator.
1. Iterator
2. List Iterator
3. Enumeration

It is only applicable for vector

1. Iterator → Iterator () : Iterator (Iterable)
 - a) iterator() method is available in Collection so that it is applicable on all child classes of Collection (ArrayList, Vector etc)
 - b) Iterator moves in forward direction only
 - c) Iterator iter = CollectionObject.iterator();

2. ListIterator :- `ListIterator l = list.iterator();`

a) `listIterator()` :- List Iterator method is available in list Collection so it is applicable on all Child Classes of list only (AL, LL, VCC, Stack).

b) It can move in forward and backward.

c) `listIterator()` list = listObject.listIterator();
`listIterator() i = listObject.listIterator();`

is used to iterate or traverse or extract a collection Stream object elements one by one.

q1 →	10	
	20	
	30	
	40	

Iterator it = q1.iterator();
it.hasNextLine();
ele = it.next();
S.delete();
}
}

→ important

```
Iterator <String> it = q1.iterator();
while (it.hasNext()) {
    String el = it.next();
    S.O.P(el);
}
```

List Iterator `String> lit = al.listIterator(2);`
String elem = lit.next();
you can also
S.O. (element)
possibly
}

→ For backward pointing use (has previous)

→ Difference b/w ArrayList and linkedlist

ArrayList
linked list

Dynamic Array
DA
1. Doubly linked list

2. Default size is 10
2. X

3. Frequently Retrieval
3. It is not suitable or
up-to task
4. First retrieval it is best
4. It is not best
5. For removal / Manipulation it
5. It is best.
is not recommended.

6. Random access is not available
6. Random access is not available
is not available

7. Queue
⑦ No queue

Whenever you have to transfer object from one place to another object is serializable.

Random Access } → Makes interface
Serializable (crappy interface)
 |
 | (There is no method)

Similarities

ArrayList Vector
LinkedList List

- | | |
|----|----------------------------|
| 1. | DA |
| 2. | 10 |
| 3. | Retrival |
| 4. | RA Serializable, Cloneable |

Difference

- | | |
|------------------------|------------------|
| 1. Non Synchronized | Synchronized |
| 2. Not thread safe | Thread Safe |
| 3. Reliability is fast | Slow |
| 4. Increase by 50% | Increase by 100% |

There are four Constructors in vector:-

```
vector();
vector(int);
vector(Collection);
vector(int, int);
(100, 20)
```

↑ Capacity increment

```
import java.util.HashSet;
```

Class Test

```
public class Test {
    public static void main(String[] args) {
        HashSet<Integer> hs = new HashSet<Integer>();
    }
}
```

1. HashSet doesn't preserved insertion order operation.

2. It doesn't allow duplicate data element if you give same data it will print once

HashSet

Capacity Load factor

greater than 1 is not recommended

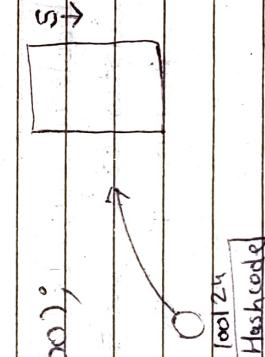
→ HashSet Capacity is 16 or default Capacity

→ LF = 0.75. → load factor

- 0.75 is an ideal
- Whenever it memory of 75% full HashSet will Create new HashSet.
- It will grow with double

Set is not a Map interface. It has 13 methods which is inherited from Collection.

HashSet Set / linked HashSet
Not give data → give data in Same Order



s.add(100);

- It is necessary to override equal with hashCode()
- When object of equal method return False hashCode will allowed entry
- How hashCode & equals method works together.
- Set maintains data uniqueness on the basis of hashCode & Equals method.

import java.util.LinkedHashSet;

```
class Customer {  
    private int id;  
    private String name;  
    private int age;  
  
    public Customer() {}  
}
```

```
public String toString() {  
    return id + " " + name + " " + age;  
}  
  
public int hashCode() {  
    return id;  
}
```

```
public boolean equals(Object o) {  
    Customer c = (Customer)o;  
    return this.id == c.id;  
}
```

↳ ClassCastException
↳ class java.lang.Comparable

Comparable → Class can only be one
Comparators → Class must me more than one

→ If you want to write your own sorting logic its mandatory
to write Comparator Class

→ These are three ways to traverse a tree.

1. NLR (Node, Leaf, Root)
2. LRN (Leaf, Root, Node)
3. RNR (Leaf, Node, Root)

→ import java.util.Comparisons;
Comparisons & Comparable

Map :-

Whenever we have to represent group of key, value pair as single unit.

Map \leftarrow -- HashMap \leftarrow LinkedHashMap

Map \leftarrow -- SortedMap \leftarrow NavigableMap \leftarrow TreeMap

Map \leftarrow -- Hashtable \leftarrow Properties

LinkedHashSet ~~&~~ Integer \rightarrow lhs = new linkedHashSet
<Integer> l();
l.add(100);

public void add (Integers, value)

→ Element method will return a set it will basically return set of entry. <100, "A"> entry single pair of

These will be interface in interface.

Class Test {

 public static void main (String args[]) {

 HashMap<Integer, String> hm = new HashMap<Integer, String>();

 hm.put(1001, "A");

 System.out.println("1. " + hm.get(1001));

 System.out.println("2. " + hm.get(1002));

 Set<Map.Entry<Integer, String>> s = hm.entrySet();

 for (Map.Entry<Integer, String> e : s) {

 System.out.println(e.getKey() + " " + e.getValue());

}

}

Que 98. What is static binding?

Ans. When type of object is determined at compile time, it is known as static binding.

Que 99. What is Dynamic binding?

Ans. When type of the object is determined at run-time, it is known as dynamic binding.