CORE JAVA

INTERVIEW QUESTIONS

YOU'LL MOST LIKELY BE ASKED

JAVA COLLECTIONS

Cheat Sheet

List	Add	Remove	Get	Contains	Next	Data Structure
ArrayList	O(1)	O(n)	O(1)	O(n)	O(1)	Array
LinkedList	O(1)	O(1)	O(n)	O(n)	O(1)	Linked List
CopyOnWriteArrayList	O(n)	O(n)	O(1)	O(n)	O(1)	Array

Set	Add	Remove	Contains	Next	Size	Data Structure
HashSet	O(1)	O(1)	O(1)	O(h/n)	O(1)	Hash Table
LinkedHashSet	O(1)	O(1)	O(1)	O(1)	O(1)	Hash Table + Linked List
EnumSet	O(1)	O(1)	O(1)	O(1)	O(1)	Bit Vector
TreeSet	O(log n)	O(log n)	O(log n)	O(log n)	O(1)	Redblack tree
CopyOnWriteArraySet	O(n)	O(n)	O(n)	O(1)	O(1)	Array
ConcurrentSkipListSet	O(log n)	O(log n)	O(log n)	O(1)	O(n)	Skip List

Map	Put	Remove	Get	ContainsKey	Next	Data Structure
HashMap	O(1)	O(1)	O(1)	O(1)	O(h / n)	Hash Table
LinkedHashMap	O(1)	O(1)	O(1)	O(1)	O(1)	Hash Table + Linked List
IdentityHashMap	O(1)	O(1)	O(1)	O(1)	O(h / n)	Array
WeakHashMap	O(1)	O(1)	O(1)	O(1)	O(h / n)	Hash Table
EnumMap	O(1)	O(1)	O(1)	O(1)	O(1)	Array
TreeMap	O(log n)	O(log n)	O(log n)	O(log n)	O(log n)	Redblack tree
ConcurrentHashMap	O(1)	O(1)	O(1)	O(1)	O(h / n)	Hash Tables
ConcurrentSkipListMap	O(log n)	O(log n)	O(log n)	O(log n)	O(1)	Skip List

Queue	Offer	Peak	Poll	Remove	Size	Data Structure
PriorityQueue	O(log n)	O(1)	O(log n)	O(n)	O(1)	Priority Heap
LinkedList	O(1)	O(1)	O(1)	O(1)	O(1)	Array
ArrayDequeue	O(1)	O(1)	O(1)	O(n)	O(1)	Linked List
ConcurrentLinkedQueue	O(1)	O(1)	O(1)	O(n)	O(1)	Linked List
ArrayBlockingQueue	O(1)	O(1)	O(1)	O(n)	O(1)	Array
PrioriritыBlockingQueue	O(log n)	O(1)	O(log n)	O(n)	O(1)	Priority Heap
SynchronousQueue	O(1)	O(1)	O(1)	O(n)	O(1)	None
DelayQueue	O(log n)	O(1)	O(log n)	O(n)	O(1)	Priority Heap
LinkedBlockingQueue	O(1)	O(1)	O(1)	O(n)	O(1)	Linked List

1) what are static blocks and static initializers in Java ?

Static blocks or static initializers are used to initialize static fields in java. we declare static blocks when we want to intialize static fields in our class. Static blocks gets executed exactly once when the class is loaded . Static blocks are executed even before the constructors are executed.

2) How to call one constructor from the other constructor ?

With in the same class if we want to call one constructor from other we use this() method. Based on the number of parameters we pass appropriate this() method is called.
Restrictions for using this method :
1) this must be the first statement in the constructor
2)we cannot use two this() methods in the constructor

3) What is method overriding in java ?

If we have methods with same signature (same name, same signature, same return type) in super class and subclass then we say
subclass method is overridden by superclass.
When to use overriding in java
If we want same method with different behaviour in superclass and subclass then we go for overriding.
When we call overridden method with subclass reference subclass method is called hiding the superclass method.

4) What is super keyword in java ?

Variables and methods of super class can be overridden in subclass . In case of overriding , a subclass object call its own variables and methods. Subclass cannot access the variables and methods of superclass because the overridden variables or methods hides the methods and variables of super class. But still java provides a way to access super class members even if its members are overridden. Super is used to access superclass variables, methods, constructors.
Super can be used in two forms :
1) First form is for calling super class constructor.
2) Second one is to call super class variables,methods.
Super if present must be the first statement.

5) Difference between method overloading and method overriding in java ?

Method Overloading	Method Overriding
1) Method Overloading occurs with in the same class	Method Overriding occurs between two classes superclass and subclass
2) Since it involves with only one class inheritance is not involved.	Since method overriding occurs between superclass and subclass inheritance is involved.
3)In overloading return type need not be the same	3) In overriding return type must be same.
4) Parameters must be different when we do overloading	4) Parameters must be same.
5) Static polymorphism can be acheived using method overloading	5) Dynamic polymorphism can be acheived using method overriding.
6) In overloading one method can't hide the another	6) In overriding subclass method hides that of the superclass method.

Overloading

```
class Dog{
    public void bark(){
        System.out.println("woof ");
    }
    //overloading method
    public void bark(int num){
        for(int i=0; i<num; i++)
            System.out.println("woof ");
    }
}
```

Same Method Name, Different Parameter

Overriding

```
class Dog{
    public void bark(){
        System.out.println("woof ");
    }
}
class Hound extends Dog{
    public void sniff(){
        System.out.println("sniff ");
    }

    public void bark(){
        System.out.println("bowl");
    }
}
```

Same Method Name, Same parameter

6) Difference between abstract class and interface ?

Interface	Abstract Class
1) Interface contains only abstract methods	1) Abstract class can contain abstract methods, concrete methods or both
2) Access Specifiers for methods in interface must be public	2) Except private we can have any access specifier for methods in abstract class.
3) Variables defined must be public , static , final	3) Except private variables can have any access specifiers
4) Multiple Inheritance in java is implemented using interface	4)We cannot achieve multiple inheritance using abstract class.
5) To implement an interface we use implements keyword	5)To implement an interface we use implements keyword

7) Why java is platform independent?

The most unique feature of java is platform independent. In any programming language soruce code is compiled in to executable code . This cannot be run across all platforms. When javac compiles a java program it generates an executable file called .class file.
class file contains byte codes. Byte codes are interpreted only by JVM's . Since these JVM's are made available across all platforms by Sun Microsystems, we can execute this byte code in any platform. Byte code generated in windows environment can also be executed in linux environment. This makes java platform independent.

8) What is method overloading in java ?

A class having two or more methods with same name but with different arguments then we say that those methods are overloaded. Static polymorphism is achieved in java using method overloading.
Method overloading is used when we want the methods to perform similar tasks but with different inputs or values. When an overloaded method is invoked java first checks the method name, and the number of arguments ,type of arguments; based on this compiler executes this method.
Compiler decides which method to call at compile time. By using overloading static polymorphism or static binding can be achieved in java.
Note : Return type is not part of method signature. we may have methods with different return types but return type alone is not sufficient to call a method in java.

9) What is difference between c++ and Java ?

Java	C++
1) Java is platform independent	C++ is platform dependent.
2) There are no pointers in java	There are pointers in C++.
3) There is no operator overloading in java	C ++ has operator overloading.
4) There is garbage collection in java	There is no garbage collection
5) Supports multithreading	Does'nt support multithreading
6) There are no templates in java	There are templates in java
7) There are no global variables in java	There are global variables in c++

10) What is JIT compiler ?

JIT compiler stands for Just in time compiler. JIT compiler compiles byte code in to executable code . JIT a part of JVM .JIT cannot convert complete java program in to executable code it converts as and when it is needed during execution.

11) What is bytecode in java ?

When a javac compiler compiler compiles a class it generates .class file. This .class file contains set of instructions called byte code. Byte code is a machine independent language and contains set of instructions which are to be executed only by JVM. JVM can understand this byte codes.

12) Difference between this() and super() in java ?

this() is used to access one constructor from another with in the same class while super() is used to access superclass constructor. Either this() or super() exists it must be the first statement in the constructor.

13) What is a class ?

Classes are fundamental or basic unit in Object Oriented Programming .A class is kind of blueprint or template for objects. Class defines variables, methods. A class tells what type of objects we are creating. For example take Department class tells us we can create department type objects. We can create any number of department objects.
All programming constructs in java reside in class. When JVM starts running it first looks for the class when we compile. Every Java application must have atleast one class and one main method.
Class starts with class keyword. A class definition must be saved in class file that has same as class name. File name must end with .java extension.

```
public class FirstClass
{public static void main(String[] args)
{System.out.println("My First class");
}
}
```

If we see the above class when we compile JVM loads the FirstClass and generates a .class file(FirstClass.class). When we run the program we are running the class and then executes the main method.

14) What is an object ?

An Object is instance of class. A class defines type of object. Each object belongs to some class.Every object contains state and behavior. State is determined by value of attributes and behavior is called method. Objects are alos called as an instance.
To instantiate the class we declare with the class type.

```
public classFirstClass {public static voidmain(String[] args)
{
FirstClass f=new FirstClass();
System.out.println("My First class");
}

}
```

To instantiate the FirstClass we use this statement
FirstClass f=new FirstClass();
f is used to refer FirstClass object.

15)What is method in java ?

It contains the executable body that can be applied to the specific object of the class.
Method includes method name, parameters or arguments and return type and a body of executable code.

```
Syntax : type methodName(Argument List){
}
```

ex : public float add(int a, int b, int c)
methods can have multiple arguments. Separate with commas when we have multiple arguments.

16) What is encapsulation ?

The process of wrapping or putting up of data in to a single unit class and keeps data safe from misuse is called encapsulation .

Through encapsulation we can hide and protect the data stored in java objects.Java supports encapsulation through access control. There are four access control modifiers in java public , private ,protected and default level.
For example take a car class , In car we have many parts which is not required for driver to know what all it consists inside. He is required to know only about how to start and stop the car. So we can expose what all are required and hide the rest by using encapsulation.

17) Why main() method is public, static and void in java ?

public : "public" is an access specifier which can be used outside the class. When main method is declared public it means it can be used outside class.
static : To call a method we require object. Sometimes it may be required to call a method without the help of object. Then we declare that method as static. JVM calls the main() method without creating object by declaring keyword static.
void : void return type is used when a method does'nt return any value . main() method does'nt return any value, so main() is declared as void.

```
Signature : public static void main(String[] args) {
```

18) Explain about main() method in java ?

Main() method is starting point of execution for all java applications.

```
public static void main(String[] args) {}
```

String args[] are array of string objects we need to pass from command line arguments.
Every Java application must have atleast one main method.

19)What is constructor in java ?

A constructor is a special method used to initialize objects in java.

we use constructors to initialize all variables in the class when an object is created. As and when an object is created it is initialized automatically with the help of constructor in java.
We have two types of constructors
Default Constructor
Parameterized Constructor
Signature : public classname()
{
}

Signature : public classname(parameters list)
{
}

20) What is difference between length and length() method in java ?

length() : In String class we have length() method which is used to return the number of characters in string.
Ex : String str = "Hello World";
System.out.println(str.length());
Str.length() will return 11 characters including space.

length : we have length instance variable in arrays which will return the number of values or objects in array.
For example :
String days[]={" Sun","Mon","wed","thu","fri","sat"};
Will return 6 since the number of values in days array is 6.

21) What is ASCII Code?

ASCII stands for American Standard code for Information Interchange. ASCII character range is 0 to 255. We can't add more characters to the ASCII Character set. ASCII character set supports only English. That

is the reason, if we see C language we can write c language only in English we can't write in other languages because it uses ASCII code.

22) What is Unicode ?

Unicode is a character set developed by Unicode Consortium. To support all languages in the world Java supports Unicode values. Unicode characters were represented by 16 bits and its character range is 0-65,535.
Java uses ASCII code for all input elements except for Strings,identifiers, and comments. If we want to use telugu we can use telugu characters for identifiers.We can enter comments in telugu.

23) Difference between Character Constant and String Constant in java ?

Character constant is enclosed in single quotes. String constants are enclosed in double quotes. Character constants are single digit or character. String Constants are collection of characters.
Ex :'2', 'A'
Ex : "Hello World"

24) What are constants and how to create constants in java?

Constants are fixed values whose values cannot be changed during the execution of program. We create constants in java using final keyword.
Ex : final int number =10;
final String str="java-interview –questions"

25) Difference between '>>' and '>>>' operators in java?

>> is a right shift operator shifts all of the bits in a value to the right to a specified number of times.
int a =15;
a= a >> 3;
The above line of code moves 15 three characters right.
>>> is an unsigned shift operator used to shift right. The places which were vacated by shift are filled with zeroes.

Core java Interview questions on Coding Standards

26) Explain Java Coding Standards for classes or Java coding conventions for classes?

Sun has created Java Coding standards or Java Coding Conventions . It is recommended highly to follow java coding standards.
Classnames should start with uppercase letter. Classnames names should be nouns. If Class name is of multiple words then the first letter of inner word must be capital letter.
Ex : Employee, EmployeeDetails, ArrayList, TreeSet, HashSet

27) Explain Java Coding standards for interfaces?

1) Interface should start with uppercase letters
2) Interfaces names should be adjectives

Example : Runnable, Serializable, Marker, Cloneable

28) Explain Java Coding standards for Methods?

1) Method names should start with small letters.
2) Method names are usually verbs
3) If method contains multiple words, every inner word should start with uppercase letter.

Ex : toString()

4) Method name must be combination of verb and noun

Ex : getCarName(),getCarNumber()

29) Explain Java Coding Standards for variables ?

1) Variable names should start with small letters.
2) Variable names should be nouns
3) Short meaningful names are recommended.
4) If there are multiple words every innerword should start with Uppecase character.

Ex : string,value,empName,empSalary

30) Explain Java Coding Standards for Constants?

Constants in java are created using static and final keywords.
1) Constants contains only uppercase letters.
2) If constant name is combination of two words it should be separated by underscore.

3) Constant names are usually nouns.
Ex:MAX_VALUE, MIN_VALUE, MAX_PRIORITY, MIN_PRIORITY

31) Difference between overriding and overloading in java?

Overriding	Overloading
In overriding method names must be same	In overloading method names must be same
Argument List must be same	Argument list must be different atleast order of arguments.
Return type can be same or we can return covariant type. From 1.5 covariant types are allowed	Return type can be different in overloading.
We cant increase the level of checked exceptions. No restrictions for unchecked exceptions	In overloading different exceptions can be thrown.
A method can only be overridden in subclass	A method can be overloaded in same class or subclass
Private,static and final variables cannot be overridden.	Private , static and final variables can be overloaded.
In overriding which method is called is decided at runtime based on the type of object referenced at run time	In overloading which method to call is decided at compile time based on reference type.
Overriding is also known as Runtime polymorphism, dynamic polymorphism or late binding	Overloading is also known as Compile time polymorphism, static polymorphism or early binding.

32) What is 'IS-A ' relationship in java?

'is a' relationship is also known as inheritance. We can implement 'is a' relationship or inheritance in java using extends keyword. The advantage or inheritance or is a relationship is reusability of code instead of duplicating the code.
Ex : Motor cycle is a vehicle
Car is a vehicle Both car and motorcycle extends vehicle.

33) What is 'HAS A'' relationship in java?

'Has a ' relationship is also known as "composition or Aggregation". As in inheritance we have 'extends' keyword we don't have any keyword to implement 'Has a' relationship in java. The main advantage of 'Has-A' relationship in java code reusability.

34) Difference between 'IS-A' and 'HAS-A' relationship in java?

IS-A relationship	HAS- A RELATIONSHIP
Is a relationship also known as inheritance	Has a relationship also known as composition or aggregation.
For IS-A relationship we uses extends keyword	For Has a relationship we use new keyword
Ex : Car is a vehicle.	Ex : Car has an engine. We cannot say Car is an engine
The main advantage of inheritance is reusability of code	The main advantage of has a relationship is reusability of code.

35) Explain about instanceof operator in java?

Instanceof operator is used to test the object is of which type.
Syntax : <reference expression> instanceof <destination type>
Instanceof returns true if reference expression is subtype of destination type.
Instanceof returns false if reference expression is null.

```
Example : public classInstanceOfExample {public static voidmain(String[] args) {Integer a =
newInteger(5);if (a instanceof java.lang.Integer) {
System.out.println(true);
} else {
System.out.println(false);
}
```

```
}

}
```

Since a is integer object it returns true.
There will be a compile time check whether reference expression is subtype of destination type. If it is not a subtype then compile time error will be shown as Incompatible types

36) What does null mean in java?

When a reference variable doesn't point to any value it is assigned null.
Example : Employee employee;
In the above example employee object is not instantiate so it is pointed no where

37) Can we have multiple classes in single file ?

Yes we can have multiple classes in single file but it people rarely do that and not recommended. We can have multiple classes in File but only one class can be made public. If we try to make two classes in File public we get following compilation error.

```
"The public type must be defined in its own file".
```

38) What all access modifiers are allowed for top class ?

For top level class only two access modifiers are allowed. public and default. If a class is declared as public it is visible everywhere.
If a class is declared default it is visible only in same package.
If we try to give private and protected as access modifier to class we get the below compilation error.

```
Illegal Modifier for the class only public,abstract and final are permitted.
```

39) What are packages in java?

Package is a mechanism to group related classes ,interfaces and enums in to a single module.
Package can be declared using the following statement :
Syntax : package <package-name>
Coding Convention : package name should be declared in small letters.
package statement defines the namespace.
The main use of package is
1) To resolve naming conflicts
2) For visibility control : We can define classes and interfaces that are not accessible outside the class.

40) Can we have more than one package statement in source file ?

We can't have more than one package statement in source file. In any java program there can be atmost only 1 package statement. We will get compilation error if we have more than one package statement in source file.

41) Can we define package statement after import statement in java?

We can't define package statement after import statement in java. package statement must be the first statement in source file. We can have comments before the package statement.

42) What are identifiers in java?

Identifiers are names in java program. Identifiers can be class name, method name or variable name.
Rules for defining identifiers in java:
1) Identifiers must start with letter,Underscore or dollar($) sign.
2) Identifiers can't start with numbers .
3) There is no limit on number of characters in identifier but not recommended to have more than 15 characters
4) Java identifiers are case sensitive.
5) First letter can be alphabet ,or underscore and dollar sign. From second letter we can have numbers .
6) We should'nt use reserve words for identifiers in java.

43) What are access modifiers in java?

The important feature of encapsulation is access control. By preventing access control we can misuse of class, methods and members.

A class, method or variable can be accessed is determined by the access modifier. There are three types of access modifiers in java. public,private,protected. If no access modifier is specified then it has a default access.

44) What is the difference between access specifiers and access modifiers in java?

In C++ we have access specifiers as public,private,protected and default and access modifiers as static, final. But there is no such divison of access specifiers and access modifiers in java. In Java we have access modifiers and non access modifiers.
Access Modifiers : public, private, protected, default
Non Access Modifiers : abstract, final, stricfp.

45) What access modifiers can be used for class ?

We can use only two access modifiers for class public and default.
public: A class with public modifier can be visible
1) In the same class
2) In the same package subclass
3) In the same package nonsubclass
4) In the different package subclass
5) In the different package non subclass.

default : A class with default modifier can be accesed
1) In the same class
2) In the same package subclass
3) In the same package nonsubclass
4) In the different package subclass
5) In the different package non subclass.

46) Explain what access modifiers can be used for methods?

We can use all access modifiers public, private,protected and default for methods.
public : When a method is declared as public it can be accessed
6) In the same class
7) In the same package subclass
8) In the same package nonsubclass
9) In the different package subclass
10) In the different package non subclass.

default : When a method is declared as default, we can access that method in
1) In the same class
2) In the same package subclass
3) In the same package non subclass
We cannot access default access method in
1) Different package subclass
2) Different package non subclass.

protected : When a method is declared as protected it can be accessed
1) With in the same class
2) With in the same package subclass
3) With in the same package non subclass
4) With in different package subclass
It cannot be accessed non subclass in different package.

private : When a method is declared as private it can be accessed only in that class.
It cannot be accessed in
1) Same package subclass
2) Same package non subclass
3) Different package subclass
4) Different package non subclass.

47) Explain what access modifiers can be used for variables?

We can use all access modifiers public, private,protected and default for variables.
public : When a variables is declared as public it can be accessed
1) In the same class

2) In the same package subclass
3) In the same package nonsubclass
4) In the different package subclass
5) In the different package non subclass.

default : When a variables is declared as default, we can access that method in
1) In the same class
2) In the same package subclass
3) In the same package non subclass
We cannot access default access variables in
4) Different package subclass
5) Different package non subclass.

protected : When a variables is declared as protected it can be accessed
1) With in the same class
2) With in the same package subclass
3) With in the same package non subclass
4) With in different package subclass
It cannot be accessed non subclass in different package.

private : When a variables is declared as private it can be accessed only in that class.
It cannot be accessed in
1) Same package subclass
2) Same package non subclass
3) Different package subclass
4) Different package non subclass.

48) What is final access modifier in java?

final access modifier can be used for class, method and variables. The main advantage of final access modifier is security no one can modify our classes, variables and methods. The main disadvantage of final access modifier is we cannot implement oops concepts in java. Ex : Inheritance, polymorphism.
final class : A final class cannot be extended or subclassed. We ar e preventing inheritance by marking a class as final. But we can still access the methods of this class by composition. Ex: String class
final methods: Method overriding is one of the important features in java. But there are situations where we may not want to use this feature. Then we declared method as final which will print overriding. To allow a method from being overridden we use final access modifier for methods.
final variables : If a variable is declared as final ,it behaves like a constant . We cannot modify the value of final variable. Any attempt to modify the final variable results in compilation error. The error is as follows
"final variable cannot be assigned."

49) Explain about abstract classes in java?

Sometimes we may come across a situation where we cannot provide implementation to all the methods in a class. We want to leave the implementation to a class that extends it. In such case we declare a class as abstract.To make a class abstract we use key word abstract. Any class that contains one or more abstract methods is declared as abstract. If we don't declare class as abstract which contains abstract methods we get compile time error. We get the following error.
"The type <class-name> must be an abstract class to define abstract methods."
Signature ; abstract class <class-name>
{
}
For example if we take a vehicle class we cannot provide implementation to it because there may be two wheelers , four wheelers etc. At that moment we make vehicle class abstract. All the common features of vehicles are declared as abstract methods in vehicle class. Any class which extends vehicle will provide its method implementation. It's the responsibility of subclass to provide implementation.
The important features of abstract classes are :
1) Abstract classes cannot be instantiated.
2) An abstract classes contains abstract methods, concrete methods or both.
3) Any class which extends abstract class must override all methods of abstract class.
4) An abstract class can contain either 0 or more abstract methods.

Though we cannot instantiate abstract classes we can create object references . Through superclass references we can point to subclass.

50) Can we create constructor in abstract class ?

We can create constructor in abstract class , it does'nt give any compilation error. But when we cannot instantiate class there is no use in creating a constructor for abstract class.

51) What are abstract methods in java?

An abstract method is the method which does'nt have any body. Abstract method is declared with keyword abstract and semicolon in place of method body.
Signature : public abstract void <method name>();
Ex : public abstract void getDetails();
It is the responsibility of subclass to provide implementation to abstract method defined in abstract class.

Java Exception Handling Interview questions

52) What is an exception in java?

In java exception is an object. Exceptions are created when an abnormal situations are arised in our program. Exceptions can be created by JVM or by our application code. All Exception classes are defined in java.lang. In otherwords we can say Exception as run time error.

53) State some situations where exceptions may arise in java?

1) Accesing an element that does not exist in array.
2) Invalid conversion of number to string and string to number.
(NumberFormatException)
3) Invalid casting of class
(Class cast Exception)
4) Trying to create object for interface or abstract class
(Instantiation Exception)

54) What is Exception handling in java?

Exception handling is a mechanism what to do when some abnormal situation arises in program. When an exception is raised in program it leads to termination of program when it is not handled properly. The significance of exception handling comes here in order not to terminate a program abruptly and to continue with the rest of program normally. This can be done with help of Exception handling.

55) What is an eror in Java?

Error is the subclass of Throwable class in java. When errors are caused by our program we call that as Exception, but some times exceptions are caused due to some environment issues such as running out of memory. In such cases we can't handle the exceptions. Exceptions which cannot be recovered are called as errors in java.
Ex : Out of memory issues.

56) What are advantages of Exception handling in java?

1) Separating normal code from exception handling code to avoid abnormal termination of program.
2) Categorizing in to different types of Exceptions so that rather than handling all exceptions with Exception root class we can handle with specific exceptions. It is recommended to handle exceptions with specific Exception instead of handling with Exception root class.
3) Call stack mechanism : If a method throws an exception and it is not handled immediately, then that exception is propagated or thrown to the caller of that method. This propogation continues till it finds an appropriate exception handler ,if it finds handler it would be handled otherwise program terminates abruptly.

57) In how many ways we can do exception handling in java?

We can handle exceptions in either of the two ways :
1) By specifying try catch block where we can catch the exception.
2) Declaring a method with throws clause .

58) List out five keywords related to Exception handling ?

1) Try
2) Catch
3) throw
4) throws
5) finally

59) Explain try and catch keywords in java?

In try block we define all exception causing code. In java try and catch forms a unit. A catch block catches the exception thrown by preceding try block. Catch block cannot catch an exception thrown by another try block. If there is no exception causing code in our program or exception is not raised in our code jvm ignores the try catch block.

Syntax :

```
try
{
}
Catch(Exception e)
{
}
```

60) Can we have try block without catch block?

Each try block requires atleast one catch block or finally block. A try block without catch or finally will result in compiler error. We can skip either of catch or finally block but not both.

61) Can we have multiple catch block for a try block?

In some cases our code may throw more than one exception. In such case we can specify two or more catch clauses, each catch handling different type of exception. When an exception is thrown jvm checks each catch statement in order and the first one which matches the type of exception is execution and remaining catch blocks are skipped.

Try with multiple catch blocks is highly recommended in java.

If try with multiple catch blocks are present the order of catch blocks is very important and the order should be from child to parent.

62) Explain importance of finally block in java?

Finally block is used for cleaning up of resources such as closing connections, sockets etc. if try block executes with no exceptions then finally is called after try block without executing catch block. If there is exception thrown in try block finally block executes immediately after catch block.

If an exception is thrown,finally block will be executed even if the no catch block handles the exception.

63) Can we have any code between try and catch blocks?

We shouldn't declare any code between try and catch block. Catch block should immediately start after try block.

```
try{
//code
}
System.out.println("one line of code"); // illegal
catch(Exception e){
//
}
```

64) Can we have any code between try and finally blocks?

We shouldn't declare any code between try and finally block. finally block should immediately start after catch block.If there is no catch block it should immediately start after try block.

```
try{
//code
}
System.out.println("one line of code"); // illegal
finally{
//
}
```

65) Can we catch more than one exception in single catch block?

From Java 7, we can catch more than one exception with single catch block. This type of handling reduces the code duplication.

Note : When we catch more than one exception in single catch block , catch parameter is implicity final. We cannot assign any value to catch parameter.

```
Ex : catch(ArrayIndexOutOfBoundsException || ArithmeticException e)
{
```

}
In the above example e is final we cannot assign any value or modify e in catch statement.

66) What are checked Exceptions?

1) All the subclasses of Throwable class except error,Runtime Exception and its subclasses are checked exceptions.
2) Checked exception should be thrown with keyword throws or should be provided try catch block, else the program would not compile. We do get compilation error.

Examples :
1) IOException,
2) SQlException,
3) FileNotFoundException,
4) InvocationTargetException,
5) CloneNotSupportedException
6) ClassNotFoundException
7) InstantiationException

67) What are unchecked exceptions in java?

All subclasses of RuntimeException are called unchecked exceptions. These are unchecked exceptions because compiler does not checks if a method handles or throws exceptions.
Program compiles even if we do not catch the exception or throws the exception.
If an exception occurs in the program,program terminates . It is difficult to handle these exceptions because there may be many places causing exceptions.
Example : 1) Arithmetic Exception
3) ArrayIndexOutOfBoundsException
4) ClassCastException
5) IndexOutOfBoundException
6) NullPointerException
7) NumberFormatException
8) StringIndexOutOfBounds
9) UnsupportedOperationException

68) Explain differences between checked and Unchecked exceptions in java?

Unchecked Exception	Checked Exception
1) All the subclasses of RuntimeException are called unchecked exception.	All subclasses of Throwable class except RuntimeException are called as checked exceptions
2) Unchecked exceptions need not be handled at compile time	Checked Exceptions need to be handled at compile time.
3) These exceptions arise mostly due to coding mistakes in our program.	
4) ArrayIndexOutOfBoundsException, ClassCastException, IndexOutOfBoundException	SqlException, FileNotFoundException,ClassNotFoundException

69) What is default Exception handling in java?

When JVM detects exception causing code, it constructs a new exception handling object by including the following information.
1) Name of Exception
2) Description about the Exception
3) Location of Exception.

After creation of object by JVM it checks whether there is exception handling code or not. If there is exception handling code then exception handles and continues the program. If there is no exception handling code JVM give the responsibility of exception handling to default handler and terminates abruptly.
Default Exception handler displays description of exception,prints the stacktrace and location of exception and terminates the program.
Note : The main disadvantage of this default exception handling is program terminates abruptly.

70) Explain throw keyword in java?

Generally JVM throws the exception and we handle the exceptions by using try catch block. But there are situations where we have to throw userdefined exceptions or runtime exceptions. In such case we use throw keyword to throw exception explicitly.
Syntax : throw throwableInstance;
Throwable instance must be of type throwable or any of its subclasses.
After the throw statement execution stops and subsequent statements are not executed. Once exception object is thrown JVM checks is there any catch block to handle the exception. If not then the next catch statement till it finds the appropriate handler. If appropriate handler is not found ,then default exception handler halts the program and prints the description and location of exception.
In general we use throw keyword for throwing userdefined or customized exception.

71) Can we write any code after throw statement?

After throw statement jvm stop execution and subsequent statements are not executed. If we try to write any statement after throw we do get compile time error saying unreachable code.

72) Explain importance of throws keyword in java?

Throws statement is used at the end of method signature to indicate that an exception of a given type may be thrown from the method.
The main purpose of throws keyword is to delegate responsibility of exception handling to the caller methods, in the case of checked exception.
In the case of unchecked exceptions, it is not required to use throws keyword.
We can use throws keyword only for throwable types otherwise compile time error saying incompatible types.
An error is unchecked , it is not required to handle by try catch or by throws.
Syntax : Class Test{
Public static void main(String args[]) throws IE
{
}
}
Note : The method should throw only checked exceptions and subclasses of checked exceptions.
It is not recommended to specify exception superclasses in the throws class when the actual exceptions thrown in the method are instances of their subclass.

73) Explain the importance of finally over return statement?

finally block is more important than return statement when both are present in a program. For example if there is any return statement present inside try or catch block , and finally block is also present first finally statement will be executed and then return statement will be considered.

74) Explain a situation where finally block will not be executed?

Finally block will not be executed whenever jvm shutdowns. If we use system.exit(0) in try statement finally block if present will not be executed.

75) Can we use catch statement for checked exceptions?

If there is no chance of raising an exception in our code then we can't declare catch block for handling checked exceptions .This raises compile time error if we try to handle checked exceptions when there is no possibility of causing exception.

76) What are user defined exceptions?

To create customized error messages we use userdefined exceptions. We can create user defined exceptions as checked or unchecked exceptions.
We can create user defined exceptions that extend Exception class or subclasses of checked exceptions so that userdefined exception becomes checked.
Userdefined exceptions can extend RuntimeException to create userdefined unchecked exceptions.
Note : It is recommended to keep our customized exception class as unchecked,i.e we need to extend Runtime Exception class but not Excpetion class.

77) Can we rethrow the same exception from catch handler?

Yes we can rethrow the same exception from our catch handler. If we want to rethrow checked exception from a catch block we need to declare that exception.

78) Can we nested try statements in java?

Yes try statements can be nested. We can declare try statements inside the block of another try statement.

79) Explain the importance of throwable class and its methods?

Throwable class is the root class for Exceptions. All exceptions are derived from this throwable class. The two main subclasses of Throwable are Exception and Error. The three methods defined in throwable class are :

1) void printStackTrace() :

This prints the exception information in the following format :

Name of the exception, description followed by stack trace.

2) getMessage()

This method prints only the description of Exception.

3) toString():

It prints the name and description of Exception.

80) Explain when ClassNotFoundException will be raised ?

When JVM tries to load a class by its string name, and couldn't able to find the class classNotFoundException will be thrown. An example for this exception is when class name is misspelled and when we try to load the class by string name hence class cannot be found which raises ClassNotFoundException.

81) Explain when NoClassDefFoundError will be raised ?

This error is thrown when JVM tries to load the class but no definition for that class is found NoClassDefFoundError will occur. The class may exist at compile time but unable to find at runtime. This might be due to misspelled classname at command line, or classpath is not specified properly , or the class file with byte code is no longer available.

Java Interview questions on threads

83) What is process ?

A process is a program in execution.

Every process have their own memory space.Process are heavy weight and requires their own address space. One or more threads make a process.

84) What is thread in java?

Thread is separate path of execution in program.

Threads are

1) Light weight
2) They share the same address space.
3) creating thread is simple when compared to process because creating thread requires less resources when compared to process
4) Threads exists in process. A process have atleast one thread.

85) Difference between process and thread?

Process	Thread
1) Program in execution.	Separate path of execution in program. One or more threads is called as process.
2) Processes are heavy weight	Threads are light weight.
3) Processes require separate address space.	Threads share same address space.
4) Interprocess communication is expensive.	Interthread communication is less expensive compared to processes.
5) Context switching from one process to another is costly.	Context switching between threads is low cost.

86) What is multitasking ?

Multitasking means ***performing more than one activity at a time*** on the computer. Example Using spreadsheet and using calculator at same time.

87) What are different types of multitasking?

There are two different types of multitasking :

1) Process based multitasking
2) Thread based multitasking

Process based multitasking : It allows to ***run two or more programs concurrently***. In process based multitasking a process is the smallest part of code .
Example : Running Ms word and Ms powerpoint at a time.

Thread based multitasking : It allows to ***run parts of a program to run concurrently***.
Example : Formatting the text and printing word document at same time .
Java supports thread based multitasking and provides built in support for multithreading.

88) What are the benefits of multithreaded programming?

Multithreading enables to use idle time of cpu to another thread which results in faster execution of program. In single threaded environment each task has to be completed before proceeding to next task making cpu idle.

89) Explain thread in java?

1) Thread is independent path of execution with in a program.
2) A thread consists of three parts Virtual Cpu, Code and data.
3) At run time threads share code and data i.e they use same address space.
4) Every thread in java is an object of java.lang.Thread class.

90) List Java API that supports threads?

java.lang.Thread : This is one of the way to create a thread. By extending Thread class and overriding run() we can create thread in java.
java.lang.Runnable : Runnable is an interface in java. By implementing runnable interface and overriding run() we can create thread in java.
java.lang.Object : Object class is the super class for all the classes in java. In object class we have three methods wait(), notify(), notifyAll() that supports threads.
java.util.concurrent : This package has classes and interfaces that supports concurrent programming.
Ex : Executor interface, Future task class etc.

91) Explain about main thread in java?

Main thread is the first thread that starts immediately after a program is started.
Main thread is important because :
1) All the child threads spawn from main thread.
2) Main method is the last thread to finish execution.

When JVM calls main method() it starts a new thread. Main() method is temporarily stopped while the new thread starts running.

92) In how many ways we can create threads in java?

We can create threads in java by any of the two ways :
1) By ***extending Thread*** class
2) By ***Implementing Runnable*** interface.

93) Explain creating threads by implementing Runnable class?

This is first and foremost way to create threads . By implementing runnable interface and implementing run() method we can create new thread.
Method signature : public void run()
Run is the starting point for execution for another thread within our program.
Example :

```
public class MyClass implements Runnable {
@Override
public void run() {
// T
}
}
```

94) Explain creating threads by extending Thread class ?

We can create a thread by extending Thread class. The class which extends Thread class must override the run() method.
Example :

```
public class MyClass extends Thread {
@Override
public void run() {
```

```
// Starting point of Execution
}
}
```

95) Which is the best approach for creating thread ?

The best way for creating threads is to implement runnable interface.
When we extend Thread class we can't extend any other class.
When we create thread by implementing runnable interface we can implement Runnable interface. In both ways we have to implement run() method.

96) Explain the importance of thread scheduler in java?

Thread scheduler is part of JVM use to determine which thread to run at this moment when there are multiple threads. Only threads in runnable state are choosen by scheduler.
Thread scheduler first allocates the processor time to the higher priority threads. To allocate microprocessor time in between the threads of the same priority, thread scheduler follows round robin fasion.

97) Explain the life cycle of thread?

A thread can be in any of the five states :
1) New : When the instance of thread is created it will be in New state.
Ex : Thread t= new Thread();
In the above example t is in new state. The thread is created but not in active state to make it active we need to call start() method on it.
2) Runnable state : A thread can be in the runnable state in either of the following two ways :
a) When the start method is invoked or
b) A thread can also be in runnable state after coming back from blocked or sleeping or waiting state.
3) Running state : If thread scheduler allocates cpu time, then the thread will be in running state.
4) Waited /Blocking/Sleeping state:
In this state the thread can be made temporarily inactive for a short period of time. A thread can be in the above state in any of the following ways:
1) The thread waits to acquire lock of an object.
2) The thread waits for another thread to complete.
3) The thread waits for notification of other thread.

5) Dead State : A thread is in dead state when thread's run method execution is complete. It dies automatically when thread's run method execution is completed and the thread object will be garbage collected.

98) Can we restart a dead thread in java?

If we try to restart a dead thread by using start method we will get run time exception since the thread is not alive.

99) Can one thread block the other thread?

No one thread cannot block the other thread in java. It can block the current thread that is running.

100) Can we restart a thread already started in java?

A thread can be started in java using start() method in java. If we call start method second time once it is started it will cause RunTimeException(IllegalThreadStateException). A runnable thread cannot be restarted.

101) What happens if we don't override run method ?

If we don't override run method .Then default implementation of Thread class run() method will be executed and hence the thread will never be in runnable state.

102) Can we overload run() method in java?

We can overload run method but Thread class start method will always cal run method with no arguments. But the overloaded method will not be called by start method we have to explicitly call this start() method.

105) What is a lock or purpose of locks in java?

Lock also called monitor is used to prevent access to a shared resource by multiple threads.
A lock is associated to shared resource. Whenever a thread wants to access a shared resource if must first acquire a lock . If already a lock has been acquired by other it can't access that shared resource. At this

moment the thread has to wait until another thread releases the lock on shared resource. To lock an object we use synchronization in java.
A lock protects section of code allowing only one thread to execute at at a time.

106) In how many ways we can do synchronization in java?

There are two ways to do synchronization in java:
1) Synchronized methods
2) Synchronized blocks
To do synchronization we use synchronize keyword.

107) What are synchronized methods ?

If we want a method of object to be accessed by single thread at a time we declare that method with synchronized keyword.
Signature :

```
public synchronized void methodName(){}
```

To execute synchronized method first lock has to be acquired on that object. Once synchronized method is called lock will be automatically acquired on that method when no other thread has lock on that method. once lock has been acquired then synchronized method gets executed. Once synchronized method execution completes automatically lock will be released. The prerequisite to execute a synchronized method is to acquire lock before method execution. If there is a lock already acquired by any other thread it waits till the other thread completes.

108) When do we use synchronized methods in java?

If multiple threads tries to access a method where method can manipulate the state of object , in such scenario we can declare a method as synchronized.

109) When a thread is executing synchronized methods , then is it possible to execute other synchronized methods simultaneously by other threads?

No it is not possible to execute synchronized methods by other threads when a thread is inside a synchronized method.

110) When a thread is executing a synchronized method , then is it possible for the same thread to access other synchronized methods of an object ?

Yes it is possible for thread executing a synchronized method to execute another synchronized method of an object.

```
public synchronized void methodName()

{
}
```

To execute synchronized method first lock has to be acquired on that object. Once synchronized method is called lock will be automatically acquired on that method when no other thread has lock on that method. once lock has been acquired then synchronized method gets executed. Once synchronized method execution completes automatically lock will be released. The prerequisite to execute a synchronized method is to acquire lock before method execution. If there is a lock already acquired by any other thread it waits till the other thread completes.

111) What are synchronized blocks in java?

Synchronizing few lines of code rather than complete method with the help of synchronized keyword are called synchronized blocks.
Signature :

```
Synchronized (object reference){// code}
```

112) When do we use synchronized blocks and advantages of using synchronized blocks?

If very few lines of code requires synchronization then it is recommended to use synchronized blocks. The main advantage of synchronized blocks over synchronized methods is it reduces the waiting time of threads and improves performance of the system.

113) What is class level lock ?

Acquiring lock on the class instance rather than object of the class is called class level lock. The difference between class level lock and object level lock is in class level lock lock is acquired on class .class instance and in object level lock ,lock is acquired on object of class.

114) Can we synchronize static methods in java?

Every class in java has a unique lock associated with it. If a thread wants to execute static synchronize method it need to acquire first class level lock. When a thread was executing static synchronized method no other thread can execute static synchronized method of class since lock is acquired on class.
But it can execute the following methods simultaneously :

1) Normal static methods
2) Normal instance methods
3) synchronize instance methods

Signature :

```
synchronized(Classname.class){}
```

115) Can we use synchronized block for primitives?

Synchronized blocks are applicable only for objects if we try to use synchronized blocks for primitives we get compile time error.

116) What are thread priorities and importance of thread priorities in java?

When there are several threads in waiting, thread priorities determine which thread to run. In java programming language every thread has a priority. A thread inherits priority of its parent thread. By default thread has normal priority of 5. Thread scheduler uses thread priorities to decide when each thread is allowed to run. Thread scheduler runs higher priority threads first.

117) Explain different types of thread priorities ?

Every thread in java has priorities in between 1 to 10. By default priority is 5 (Thread.NORM_PRIORITY). The maximum priority would be 10 and minimum would be 1.Thread class defines the following constants(static final variables) to define properties.
Thread. MIN_PRIORITY = 1;
Thread.NORM_PRIORITY=5;
Thread. MAX_PRIORITY=10;

118) How to change the priority of thread or how to set priority of thread?

Thread class has a set method to set the priority of thread and get method to get the priority of the thread.
Signature : final void setPriority(int value);
The setPriority() method is a request to jvm to set the priority. JVM may or may not oblige the request.
We can get the priority of current thread by using getPriority() method of Thread class.

```
final int getPriority()
{
}
```

119) If two threads have same priority which thread will be executed first ?

We are not guaranteed which thread will be executed first when there are threads with equal priorities in the pool. It depends on thread scheduler to which thread to execute. The scheduler can do any of the following things :

1) It can pick any thread from the pool and run it till it completes.
2) It can give equal opportunity for all the threads by time slicing.

120) What all methods are used to prevent thread execution ?

There are three methods in Thread class which prevents execution of thread.

1) yield()
2) join()
3) sleep()

121) Explain yield() method in thread class ?

Yield() method makes the current running thread to move in to runnable state from running state giving chance to remaining threads of equal priority which are in waiting state. yield() makes current thread to sleep for a specified amount of time.There is no guarantee that moving a current running thread from runnable to running state. It all depends on thread scheduler it doesn't gurantee anything.

Calling yield() method on thread does not have any affect if object has a lock. The thread does'nt lose any lock if it has acquired a lock earlier.

Signature :

```
public static native void yield()
{

}
```

122) Is it possible for yielded thread to get chance for its execution again ?

Yield() causes current thread to sleep for specified amount of time giving opportunity for other threads of equal priority to execute. Thread scheduler decides whether it get chance for execution again or not. It all depends on mercy of thread scheduler.

123) Explain the importance of join() method in thread class?

A thread can invoke the join() method on other thread to wait for other thread to complete its execution. Assume we have two threads, t1 and t2 threads . A running thread t1 invokes join() on thread t2 then t1 thread will wait in to waiting state until t2 completes. Once t2 completes the execution, t1 will continue.
join() method throws Interrupted Exception so when ever we use join() method we should handle Interrrupted Exception by throws or by using try catch block.
Signature :

```
public final void join() throws InterruptedException
{

}
```

```
public final synchronized void join(long millis)
throws InterruptedException
{

}
```

```
public final synchronized void join(long millis, int nanos)
throws InterruptedException
{

}
```

124) Explain purpose of sleep() method in java?

sleep() method causes current running thread to sleep for specified amount of time . sleep() method is the minimum amount of the time the current thread sleeps but not the exact amount of time.

Signature :

```
  public static native void sleep(long millis) throws InterruptedException
{

}
```

```
public static void sleep(long millis, int nanos)
throws InterruptedException {


}
```

125) Assume a thread has lock on it, calling sleep() method on that thread will release the lock?

Calling sleep() method on thread which has lock does'nt affect. Lock will not be released though the thread sleeps for a specified amount of time.

126) Can sleep() method causes another thread to sleep?

No sleep() method causes current thread to sleep not any other thread.

127) Explain about interrupt() method of thread class ?

Thread class interrupt() method is used to interrupt current thread or another thread. It doesnot mean the current thread to stop immediately, it is polite way of telling or requesting to continue your present work. That is the reason we may not see the impact of interrupt call immediately.
Initially thread has a boolean property(interrupted status) false. So when we call interrupt() method status would set to true. This causes the current thread to continue its work and does not have impact immediately.
If a thread is in sleeping or waiting status (i.e thread has executed wait () or sleep() method) thread gets interrupted it stops what it is doing and throws an interrupted exception. This is reason we need to handle interrupted exception with throws or try/ catch block.

128) Explain about interthread communication and how it takes place in java?

Usually threads are created to perform different unrelated tasks but there may be situations where they may perform related tasks. Interthread communication in java is done with the help of following three methods :

1) wait()
2) notify()
3) notifyAll()

129) Explain wait(), notify() and notifyAll() methods of object class ?

wait() : wait() method() makes the thread current thread sleeps and releases the lock until some other thread acquires the lock and calls notify().
notify() :notify() method wakes up the thread that called wait on the same object.
notfiyAll() :notifyAll() method wakes up all the threads that are called wait() on the same object. The highest priority threads will run first.
All the above three methods are in object class and are called only in synchronized context.
All the above three methods must handle InterruptedException by using throws clause or by using try catch clause.

130) Explain why wait() , notify() and notifyAll() methods are in Object class rather than in thread class?

First to know why they are in object class we should know what wait(), notify(), notifyAll() methods do. wait() , notify(), notifyAll() methods are object level methods they are called on same object.wait(), notify(), notifyAll() are called on an shared object so to they are kept in object class rather than thread class.

131) Explain IllegalMonitorStateException and when it will be thrown?

IllegalMonitorStateException is thrown when wait(), notify() and notifyAll() are called in non synchronized context. Wait(), notify(),notifyAll() must always be called in synchronized context other wise we get this run time exception.

132) when wait(), notify(), notifyAll() methods are called does it releases the lock or holds the acquired lock?

wait(), notify(), notifyAll() methods are always called in synchronized context. When these methods are called in synchronized context.
So when they enter first in synchronized context thread acquires the lock on current object. When wait(), notify(), notifyAll() methods are called lock is released on that object.

133) Explain which of the following methods releases the lock when yield(), join(),sleep(),wait(),notify(), notifyAll() methods are executed?

Method	Releases lock (Yes or No)
yield()	No
sleep()	No
join()	No
wait()	Yes

Notify()	Yes
notifyAll()	Yes

134) What are thread groups?

Thread Groups are group of threads and other thread groups. It is a way of grouping threads so that actions can be performed on set of threads for easy maintenance and security purposes.
For example we can start and stop all thread groups. We rarely use thread group class. By default all the threads that are created belong to default thread group of the main thread. Every thread belongs to a thread group. Threads that belong to a particular thread group cannot modify threads belonging to another thread group.

135) What are thread local variables ?

Thread local variables are variables associated to a particular thread rather than object. We declare ThreadLocal object as private static variable in a class. Everytime a new thread accesses object by using getter or setter we are accesing copy of object. Whenever a thread calls get or set method of ThreadLocal instance a new copy is associated with particular object.

136) What are daemon threads in java?

Daemon threads are threads which run in background. These are service threads and works for the benefit of other threads. Garbage collector is one of the good example for daemon threads.
By default all threads are non daemon. Daemon nature of a thread can be inherited. If parent thread is daemon , child thread also inherits daemon nature of thread.

137) How to make a non daemon thread as daemon?

By default all threads are non daemon. We can make non daemon nature of thread to daemon by using setDaemon() method. The important point to note here we can call setDaemon() only before start() method is called on it. If we call setDaemon() after start() method an IllegalThreadStateException will be thrown.

138) Can we make main() thread as daemon?

Main thread is always non daemon. We cannot change the non daemon nature of main thread to daemon.

Interview questions on Nested classses and inner classes

139) What are nested classes in java?

Class declared with in another class is defined as nested class.
There are two types of nested classes in java.
1) Static nested class
2) Non static nested class
A static nested class has static keyword declared before class definition.

140) What are inner classes or non static nested classes in java?

Nested classes without any static keyword declaration in class definition are defined as non static nested classes. Generally non static nested classes are referred as inner classes.
There are three types of inner classes in java :
1) Local inner class
2) Member inner class
3) Anonymous inner class

141) Why to use nested classes in java?
(or)
What is the purpose of nested class in java?

1) Grouping of related classes

Classes which are not reusable can be defined as inner class instead of creating inner class.
For example : We have a submit button upon click of submit button we need to execute some code. This code is related only to that class and cannot be reused for other class . Instead of creating a new class we can create inner class

2) To increase encapsulation :

Inner class can access private members of outer class.so by creating getter and setter methods for private variables , outside world can access these variables. But by creating inner class private variables can be accesed only by inner class.

3) Code readable and maintainable :

Rather than creating a new class we can create inner class so that it is easy to maintain.

4) Hiding implementation :
Inner class helps us to hide implementation of class.

142) Explain about static nested classes in java?
When a static class is defined inside a enclosing class we define that as nested class. Static nested classes are not inner classes. Static nested classes can be instantiated without instance of outer class.
A static nested doesnot have access to instance variables and non static methods of outer class.

143) How to instantiate static nested classes in java?
We can access static members and static methods of outer class without creating any instance of outer class.
Syntax for instantiating Static nested class :
OuterclassName.StaticNestedClassName ref=new OuterclassName.StaticNestedClassName();

144) Explain about method local inner classes or local inner classes in java?
Nested classes defined inside a method are local inner classes. We can create objects of local inner class only inside method where class is defined. A local inner classes exist only when method is invoked and goes out of scope when method returns.

145) Explain about features of local inner class?
1) Local inner class does not have any access specifier.
2) We cannot use access modifiers static for local inner class. But we can use abstract and final for local inner class.
3) We cannot declare static members inside local inner classes.
4) We can create objects of local inner class only inside method where class is defined.
5) Method local inner classes can only access final variables declared inside a method.
6) Method local inner classes can be defined inside loops(for,while) and blocks such as if etc.

146) Explain about anonymous inner classes in java?
Inner class defined without any class name is called anonymous inner class. Inner class is declared and instantiated using new keyword.The main purpose of anonymous inner classes in java are to provide interface implementation. We use anonymous classes when we need only one instance for a class. We can use all members of enclosing class and final local variables.
When we compile anonymous inner classes compiler creates two files
1) EnclosingName.class
2) EnclsoingName$1.class

147) Explain restrictions for using anonymous inner classes?
1) An anonymous inner class cannot have any constructor because there is no name for class.
2) An anonymous inner class cannot define static methods, fields or classes.
3) We cannot define an interface anonymously.
4) Anonymous inner class can be instantiated only once.

148) Is this valid in java ? can we instantiate interface in java?
```
Runnable r = new Runnable() {
@Override
public void run() {
}
};
```
Runnable is an interface.If we see the above code it looks like we are instantiating Runnable interface. But we are not instantiating interface we are instantiating anonymous inner class which is implementation of Runnable interface.

149) Explain about member inner classes?
Non static class defined with in enclosing class are called member inner class. A member inner class is defined at member level of class. A member inner class can access the members of outer class including private members.
Features of member inner classes :
1) A member inner class can be declared abstract or final.
2) A member inner class can extend class or implement interface.
3) An inner class cannot declare static fields or methods.
4) A member inner class can be declared with public, private, protected or default access.

150) How to instantiate member inner class?

OuterClassName.InnerclassName inner=new OuterClassReference.new InnerClassName();
We cannot instantiate inner class without outer class reference

151) How to do encapsulation in Java?

Make instance variables private.
Define getter and setter methods to access instance variables .

152) What are reference variables in java?

Variables which are used to access objects in java are called reference variables.
Ex : Employee emp=new Employee();
In the above example emp is reference variable.
Reference variable can be of only one type.
A reference variable can point to any number of objects. But if a reference variable is declared final it can't point to other objects.
A reference variable can be declared either to a class type or interface type. If a reference variable is declared with interface type it points to the class that implements the interface.

153) Will the compiler creates a default constructor if I have a parameterized constructor in the class?

No compiler won't create default constructor if there is parameterized constructor in the class. For example if I have a class with no constructors, then compiler will create default constructor.
For Example :

```
public classCar {}
```

In the above Car class there are no constructors so compiler creates a default constructor.

```
public classCar {Car(String name) {

}

}
```

In this example compiler won't create any default constructor because already there is one constructor in the Car class.

154) Can we have a method name same as class name in java?

Yes we can have method name same as class name it won't throw any compilation error but it shows a warning message that method name is same as class name.

155) Can we override constructors in java?

Only methods can be overridden in java. Constructors can't be inherited in java. So there is no point of verriding constructors in java.

156) Can Static methods access instance variables in java?

No.Instance variables can't be accessed in static methods. When we try to access instance variable in static method we get compilation error. The error is as follows:

```
Cannot make a static reference to the non static field name
```

157) How do we access static members in java?

Instance variables and instance methods can be accessed using reference variable . But to access static variables or static methods we use Class name in java.

158) Can we override static methods in java?

Static methods can't be overridden. If we have a static method in superclass and subclass with same signature then we don't say that as overriding. We call that as

159) Difference between object and reference?

Reference and object are both different. Objects are instances of class that resides in heap memory. Objects does'nt have any name so to access objects we use references. There is no alternative way to access objects except through references.

Object cannot be assigned to other object and object cannot be passed as an argument to a method. Reference is a variable which is used to access contents of an object. A reference can be assigned to other reference ,passed to a method.

160) Objects or references which of them gets garbage collected?

Objects get garbage collected not its references.

161) How many times finalize method will be invoked ? who invokes finalize() method in java?

Finalize () method will be called only once on object. Before the object gets garbage collected garbage collector will call finalize() method to free the resources. Finalize() method will be called only when object is eligible for garbage collection.

162) Can we able to pass objects as an arguments in java?

Only references can be passed to a method not an object. We cannot pass the objects to a method. The largest amount of data that can passed as parameters are long or double.

163) Explain wrapper classes in java?

Converting primitives to objects can be done with the help of wrapper classes. Prior to java 1.5 we use Wrapper classes to convert primitives to objects. From java 1.5 we have a new feature autoboxing which is used to convert automatically primitives to objects but in wrapper classes programmer has to take care of converting primitives to objects.
Wrapper classes are immutable in java. Once a value is assigned to it we cannot change the value.

164) Explain different types of wrapper classes in java?

For every primitive in java we have corresponding wrapper class. Here are list of wrapper classes available in java.

Primtive	Wrapper Class
boolean	Boolean
int	Integer
float	Float
char	Character
byte	Byte
long	Long
short	Short

165) Explain about transient variables in java?

To save the state of an object to persistent state we use serialization. If we want a field or variable in the object not to be saved, then we declare that variable or field as transient.

```
Example : public Class Car implements serializable
{
transient int carnumber;
}
```

166) Can we serialize static variables in java?

Static variables cannot be serialized in java.

167) What is type conversion in java?

Assigning a value of one type to variable of other type is called type conversion.

```
Example : int a =10;
long b=a;
```

There are two types of conversion in java:

1) Widening conversion
2) Narrowing conversion

168) Explain about Automatic type conversion in java?

Java automatic type conversion is done if the following conditions are met :

1) When two types are compatible

Ex : int, float
int can be assigned directly to float variable.

2) Destination type is larger than source type.

Ex : int, long

Int can be assigned directly to long .Automatic type conversion takes place if int is assigned to long because long is larger datatype than int.
Widening Conversion comes under Automatic type conversion.

169) Explain about narrowing conversion in java?

When destination type is smaller than source type we use narrowing conversion mechanism in java. Narrowing conversion has to be done manually if destination type is smaller than source type. To do narrowing conversion we use cast. Cast is nothing but explicit type conversion.
Example : long a;
byte b;
b=(byte)a;
Note : casting to be done only on valid types otherwise classcastexception will be thrown.

170) Explain the importance of import keyword in java?

Import keyword is used to import single class or package in to our source file.import statement is declared after package decalaration. We use wild character (*) to import package.
Note : After compilation the compiled code does not contain import statement it will be replaced with fully qualified class names

171) Explain naming conventions for packages ?

Sun defined standard naming conventions for packages.
1) Package names should be in small letters.
2) Package name starts with reverse company domain name (excluding www) followed by department and project name and then the name of package.
Example : com.google.sales.employees

172) What is classpath ?

The path where our .class files are saved is referred as classpath.JVM searches for .class files by using the class path specified. Class path is specified by using CLASSPATH environment variable. CLASSPATH environment variable can contain more than one value. CLASSPATH variable containing more than one value is separated by semicolon.
Example to set class path from command prompt :
set CLASSPATH= C:Program FilesJavajdk1.6.0_25bin;.;
only parent directories need to be added to classpath.Java compiler will look for appropriate packages and classes.

173) What is jar ?

Jar stands for java archive file. Jars are created by using Jar.exe tool. Jar files contains .class files, other resources used in our application and manifest file.Manifest file contains class name with main method.jar contains compressed .class files. Jvm finds these .class files without uncompressing this jar.

174) What is the scope or life time of instance variables ?

When object is instantiated using new operator variables get allocated in the memory.instance variables remain in memory till the instance gets garbage collected

175) Explain the scope or life time of class variables or static variables?

Static variables do not belong to instances of the class. We can access static fields even before instantiating the class. Static variable remain in memory till the life time of application.

176) Explain scope or life time of local variables in java?

Local variables are variables which are defined inside a method. When the method is created local variables gets created in stack memory and this variable gets deleted from memory once the method execution is done.

177) Explain about static imports in java?

From Java 5.0 we can import static variables in to source file. Importing static member to source file is referred as static import. The advantage of static import is we can access static variables without class or interface name.
Syntax : import static packagename.classname.staticvariablename;
Ex : import static com.abc.Employee.eno;
To import all static variables from a class in to our source file we use *.
import static com.abc.Employee.*

178) Can we define static methods inside interface?

We can't declare static methods inside interface. Only instance methods are permitted in interfaces.only public and abstract modifiers are permitted for interface methods. If we try to declare static methods inside interface we get compilation error saying
"Illegal modifier for the interface method Classname.methodName(); only public & abstract are permitted".

179) Define interface in java?

Interface is collection of abstract methods and constants. An interface is also defined as pure or 100 percent abstract class.Interfaces are implicitly abstract whether we define abstract access modifier or not. A class implementing interface overrides all the abstract methods defined in interface. Implements keyword is used to implement interface.

180) What is the purpose of interface?

Interface is a contract . Interface acts like a communication between two objects. When we are defining interface we are defining a contract what our class should do but not how it does. An interface does'nt define what a method does. The power of interface lies when different classes that are unrelated can implement interface. Interfaces are designed to support dynamic method resolution at run time.

181) Explain features of interfaces in java?

1) All the methods defined in interfaces are implicitly abstract even though abstract modifier is not declared.
2) All the methods in interface are public whether they are declared as public or not.
3) variables declared inside interface are by default public, static and final.
4) Interfaces cannot be instantiated.
5) we cannot declare static methods inside interface.
6) ` implements' keyword is used to implement interface.
7) Unlike class, interface can extend any number of interfaces.
8) We can define a class inside interface and the class acts like inner class to interface.
9) An interface can extend a class and implement an interface
10) Multiple inheritance in java is achieved through interfaces.

182) Explain enumeration in java?

Enumeration is a new feature from Java 5.0. Enumeration is set of named constants . We use enum keyword to declare enumeration. The values defined in enumeration are enum constants.Each enum constant declared inside a enum class is by default public , static and final.
Example :

```
package javaexamples;
public enum Days {
SUN,MON,TUE,WED,THU,FRI,SAT;
}
```

SUN,MON,TUE,WED,THU,FRI,SAT are enum constants.

183) Explain restrictions on using enum?

1) Enums cannot extend any other class or enum.
2) We cannot instantiate an enum.
3) We can declare fields and methods in enum class. But these fields and methods should follow the enum constants otherwise we get compilation error.

184) Explain about field hiding in java?

If superclass and subclass have same fields subclass cannot override superclass fields. In this case subclass fields hides the super class fields. If we want to use super class variables in subclass we use super keyword to access super class variables.

185) Explain about Varargs in java?

Beginning with Java 5 has a new feature Varargs which allows methods to have variable number of arguments. It simplifies creation of methods when there are more number of arguments. Earlier to java 5 Varargs are handled by creating method with array of arguments.
Ex : public static void main(String[] args)
A variable length argument is specified using ellispses with type in signature. main method with var args is written as follows:
public static void main(String … args)
If no arguments are passes we get array with size 0.There is no need for null check if no arguments are passed.

186) Explain where variables are created in memory?

When we declare variables variables are created in stack. So when the variable is out of scope those variables get garbage collected.

187) Can we use Switch statement with Strings?

Prior to Java 7 we can use only int values and enum constants in Switch .Starting with Java 7 we can use strings in Switch statement. If we use strings in switch statement prior to Java 7 we will get compile time error "only int and enum constants are permitted".

188) In java how do we copy objects?

In Java we cannot copy two objects but by assigning one reference to other we can copy objects. For example if we have a reference r1 that point to object .so when we declare r2=r1, we are assigning reference r1 to r2 so now r2 points to the same object where r1 points. Any changes done by one reference on an object will reflect to other.

Oops concepts interview questions

189) Explain about procedural programming language or structured programming language and its features?

In traditional programming language to solve a problem we use set of procedures. Once the procedures or functions are determined next they concentrate on storing data.

Features :

1) In this top down approach is followed. First procedures were determined and then concentrate on minute details.
2) Concentrate more on functions and procedure rather than data.
3) In traditional programming language procedures manipulate global data without knowing to other procedures.
4) Very little concentration on minute details

The main drawback of traditional programming languages works well only for small problems. But not suitable for larger problems.

Ex : C language, Pascal

190) Explain about object oriented programming and its features?

Java replaced traditional programming language developed in 1970's. In Object oriented programming everything is made up of object. In this language bottom up approach is followed. Each object communicates with other as opposed to traditional view.

Features :

1) In this bottom approach is followed. First concentrates on minute details like creating objects then concentrates on implementation or solving the problem.
2) Concentrate more on data and give less importance for implementation.
3) Objects communicate with each other

The main advantage of object oriented programming language is works well for larger problems.

191) List out benefits of object oriented programming language?

1) Easy maintenance
2) Code reusability
3) Code extendability
4) Reliable

192) Differences between traditional programming language and object oriented programming language?

Traditional Programming language	Object Oriented Programming Language
A program is divided in to modules and procedures.	A program is divided in to number of objects.
Implementation is done through procedures.	Implementation is done through interfaces.
In traditional programming there is no encapsulation all procedures access data.	In oops encapsulation is done by tightly coupling data and behaviour together in class.
Suitable for small programs or problems	Suitable for large programs and complex problems.

193) Explain oops concepts in detail?

Object oriented programming should support these three features :

1) Inheritance
2) Encapsulation

3) Polymorphism

194) Explain what is encapsulation?

Encapsulation is the process of wrapping of code and behaviour in a single unit called class and preventing from misuse is called encapsulation. Encapsulation exposes only part of object which are safe to exposed and remaining part of object is kept secured.
Encapsulation is supported through access control in java. There are four types of access control specifiers(public,private, protected, default) in java which supports encapsulation.
For example tv manufacturers exposes only buttons not all the thousands of electronic components which it is made up of.

195) What is inheritance ?

Inheritance is one of the important feature of object oriented language. Inheriting is the process of acquiring features of others. For example a child acquires the features of their parents.
In java inheritance is the process of inheriting member of existing classes by extending their functionality. The original class is called base class, parent class or super class. The new class derived from parent is called child class, sub class, and derived class.
We use extends keyword in java to extend a class in java. All java classes extend java.lang.Object since object class is the super class for all classes in java.
When we create a new class by using inheritance 'is-a' relationship is formed.

196) Explain importance of inheritance in java?

Reusability :The major advantage of inheritance is code reuse. We can avoid duplicating code by using inheritance. We can place all common state and behaviour in that class , by extending that class we can
Extendability : We can add new functionality to our application without touching the existing code.
For example if we take Ms word we came across number of versions of msword such as word 2003,2007. Everytime they won't write new code they reuse the existing code and some more features.

197) What is polymorphism in java?

Polymorphism is combination of two greek words which mean many forms. In polymorphism actual type of object involved in method call determines which method to call rather type of reference variable.
59) What is covariant return ?
In java 1.4 and earlier one method can override super class method if both methods have same signature and return types.
From Java 1.5 , a method can override other method if argument types match exactly though return types are different.(Return type must be subtype of other method).
Example : Class A

```
{

A doSomeThing()
{
return new A();
}
}
```

Example : Class B

```
{

B doSomeThing()
{
return new B();
}
}
```

From java 1.5 return type for doSomeThing() in Class B is valid . We get compile time error in 1.4 and earlier.

Collection Framework interview questions

198) What is collections framework ?

A framework is set of classes and interfaces to build a functionality. Java collections framework provides set of interfaces and classes for storing and manipulating collections. Collection framework contains classes and interfaces in java.util package and java.util.concurrent packages.
Advantages or benefits of Collections framework :

1) High performance
2) Using this framework we can create different types of collections

3) We can create our own collection and we can extend a collection.
4) Reduces programming effort.
5) Increases speed and quality : Collections framework provides high performance, implementations of useful data structures and algorithms.

199) What is collection ?

A collection is a container which holds group of objects. Collection provides a way to manage objects easily. Collections manages group of objects as single unit.
Examples include list of strings, integers etc.
Here are few basic operations we do on collections :
1) Adding objects to collection.
2) Removing or deleting objects from collection.
3) Retrieving object from collection.
4) Iterating collection.

200) Difference between collection, Collection and Collections in java?

collection : represent group of objects where objects are stored.
Collection : This is one of the core interface which provides basic functionality for collection.
Collections : Collections contains some utility static methods that operate on collections.

201) Explain about Collection interface in java ?

Collection is the fundamental and root interface in Collections framework. Collection extends Iterable interface and inherits iterator method which returns Iterator object.
Signature :

```
public interface Collection<E> extends Iterable<E> {
}
```

Methods in Collection interface :

boolean add(E e);	Adds an element to the collection. Returns true if element is added.
boolean remove(Object o);	Removes an object from collection if that object is present in collection. Return true if matching object is removed from collection.
boolean addAll(Collection<? extends E> c);	Adds all the elements specified in the collection to this collection.Returns true if all elements are added.
boolean removeAll(Collection<?> c);	Removes all the elements from this collection that are specified in other collection.Returns true if all the elements are removed.
int size();	Returns number of elements in collection.
boolean isEmpty();	Checks whether collection contains elements or not. If no elements are present it returns false.
boolean contains(Object o);	Checks whether specified object is in collection or not. Return true if object is in collection.
Iterator<E> iterator();	Used to iterator over collection. No guarantee on order of elements iterated.
boolean retainAll(Collection<?> c);	Removes all the elements which are not in specified collection. Returns only elements specified in collection removing other elements.
Object[] toArray();	Returns an array of elements in collection.

202) List the interfaces which extends collection interface ?

1) List
2) Set
3) Queue
4) Deque (From Java 6)

203) Explain List interface ?

List interface extends collection interface used to store sequence of elements in collection.
We can even store duplicate elements in list.
We can insert or access elements in list by using index as we do in arrays.
List is an ordered collection.
The main difference between List and non list interface are methods based on position.

Some of the operations we can perform on List :
1) Adding an element at specified index.
2) Removing an element at specified index.
3) To get the index of element
List contains some specific methods apart from Collection interface methods.

204) Explain methods specific to List interface ?

boolean addAll(int index, Collection<? extends E> c);	This method inserts all the elements in specified collection to the list at specified position.
E get(int index);	This method returns an element at specified position in the list.
E set(int index, E element);	This method replaces the element at specified position in the list with the specified element.
void add(int index, E element);	This method inserts the specified element with the index specified.
E remove(int index);	This method removes the element at specified index and returns the element removed.
int indexOf(Object o);	indexOf() method returns the index of last occurrence of specified element. If there is no element in the list it removes the element.
ListIterator<E> listIterator();	Returns a list iterator of elements in list.
List<E> subList(int fromIndex, int toIndex);	This method returns list of elements between indexes specified.

205) List implementations of List Interface ?
1) ArrayList
2) Vector
3) LinkedList

206) Explain about ArrayList ?
ArrayList is an ordered collection which extends AbstractList and implements List interface.
We use ArrayList mainly when we need faster access and fast iteration of elements in list.
We can insert nulls in to arraylist.
Arraylist is nothing but a growable array.

```
public class ArrayList<E> extends AbstractList<E>    implements List<E>,
RandomAccess, Cloneable, java.io.Serializable{}
```

From java 1.4 ArrayList implements RandomAccess interface which is a marker interface which supports fast and random access.
Advantages :
1) Faster and easier access.
2) Used for Random access of elements.
Drawbacks :
1) We cannot insert or delete elements from middle of list.

207) Difference between Array and ArrayList ?
Arrays are used to store primitives or objects of same type or variables that are subclasses of same type.
ArrayList : It is an ordered collection which grows dynamically.
In list we can insert nulls values and list allows duplicate elements.

ARRAY	ARRAY LIST
1) While creating array we have to know the size.	1) But it is not required to know size while creating ArrayList, because arraylist grows dynamically.
2) To put an element in to array we use the following syntax :String array[] = **new**String[5];array[1] = "java";We must know specific location to insert an element in to	2) We can add element to arraylist with following syntax :List<String> stringList = new ArrayList<String>();stringList.add("java");

array. If we try to put element in index which is out of range we get ArrayIndexOutOfBounds Exception	
3) Arrays are static	3) ArrayList is dynamic
4) We can store objects and primitives	4) We can store only primitives prior to 1.5 . From 1.5 we can store even objects also.
5) We have to manually write logic for inserting and removing elements.	5) Just a method call would add or remove elements from list.
6) Arrays are faster	6) Arraylist is slower.
	7) Arraylist is implemented using arrays.

208) What is vector?

Vector is similar to arraylist used for random access.
Vector is a dynamic array like arraylist.
vector size increases or decreases when elements are added and removed .
Vector is synchronized .
vector and Hashtable are the only collections since 1.0.
Rest of the collections are added from 2.0.

```
public class Vector<E>extends AbstractList<E>implements List<E>,
RandomAccess, Cloneable, java.io.Serializable
```

209) Difference between arraylist and vector ?

Both ArrayList and vector grows dynamically. The differences between arraylist and vector are :

1) Arraylist is not synchronized and vector is synchronized.
2) Vector is legacy collection introduced in 1.0 and Arraylist introduced in java 2.0.

Performance wise it is recommended to use arraylist rather than vector because by default vector is synchronized which reduces performance if only one thread accesses it.

210) Define Linked List and its features with signature ?

Linked list is used for storing a collection of objects that allows efficient addition and removal of elements in the middle of the collection.
The main drawback with arrays is if we want to insert an element in the middle of the list we need to move each element to next position and insert the element. Similarly with remove if we want to remove an element we need to remove the element and move the list of elements.
But with linked list we can insert and delete in the middle of the list efficiently by just updating the neighbouring node reference.
Linked list class is in java.util package.
Linked List class extends class extends AbstractSequentialList and I mplements List, Deque, Cloneable and Serializable.

```
Signature :public class LinkedList<E>    extends
AbstractSequentialList<E>
   implements List<E>, Deque<E>, Cloneable, java.io.Serializable
{
}
```

Important methods specific to LinkedList class :

1) public E getFirst() :

getFirst() will returns the first element in the list.

2) public E getLast() :

getLast() returns the last element in the list.

3) public E removeFirst() :

removeFirst() method removes the first element in the list.

4) public E removeLast() :

removeLast() method removes the last element in the list.

5) public void addFirst(E e)
Inserts the element at beginning of the list.

6) public void addLast(E e) :
Inserts the element at end of the list.

211) Define Iterator and methods in Iterator?

If we want to iterate through all the elements in collection we use Iterator. Iterator is a standard way to access elements one by one in collection. Iterator is an object associated with collection used to loop through the collection.
Steps for accessing elements in Iterator :
1) Obtain Iterator object by calling iterator() method on collection.

```
Ex : ArrayList <String> al=new ArrayList<String>();
Iterator itr=al.iterator();
```

2) Call hasNext() method on iterator object in loop as long as hasNext() returns true.

```
Ex : while(itr.hasNext())
{
}
```

3) Get each element by calling next() inside the loop.

```
while(itr.hasNext())
{
String str=itr.next();
}
```

Methods in iterator :

Method	Description
boolean hasNext();	This method returns true if there is next element.hasNext() points to position before first lement.If there are any elements if will return true.
E next();	Returns the next element in the iteration. . If there are no elements in the Iteration NoSuchElementException is thrown. next() will move the pointer to next position and returns the element.
void remove();	Removes the element.

Note : If we call next() on last element it will throw java.util.NoSuchElementException. So before calling next() first we should call hasNext() whether it has elements or not. If there is next element we can call next() so that we can avoid exception.

212) In which order the Iterator iterates over collection?

The order in which Iterator will iterate the collection depends on the traversal order of collection.
For example : for list traversal order will be sequential, and for set the order cannot be determined, and for sorted sorted set will sort the elements in sorted order.
So it all depends on the collection in which order iterator iterates.

212) Explain ListIterator and methods in ListIterator?

List Iterator is similar to Iterator but ListIterator is bidirectional.
We can traverse through the collection in either forward or backward direction.
List Iterator extends Iterator and all the methods in Iterator will be there in ListIterator too with some additional methods .
List Iterator doesn't have current element .Position of List Iterator lies between two elements i.e previous element and next element.
Features of ListIterator :
1) Traversal of List in either direction.
2) Modification of its elements.
3) Access to elements position.

Signature :

```
public interface ListIterator<E> extends Iterator<E> {
}
```

ListIterator methods :

Method	Description

Void add(E obj)	Inserts element in to the list infront of the element returned by call to next() and after the element returned by call to next().
boolean hasNext();	Returns true if there are more elements in the list instead of throwing exception if there are no elements.
E next();	Returns the next element . NoSuchElementException is thrown if there is no next element.
boolean hasPrevious();	Returns true if there are elements when iterating list in reverse direction.
E previous();	Returns the previous element in the list.
int nextIndex();	Returns the index of the element returned by next() method. If there are no elements it returns the size of the list.
int previousIndex();	Returns the index of the element returned by previous() method. If there are no elements it returns the size of the list. Returns -1 if the iterator is at beginning of list.
void remove();	Removes the element that was returned by calling next() or previous(). An Illegal state Exception will be thrown if remove() is called before next() or previous().
void set(E e);	This method replaces an element in the list with the specified element.

213) Explain about Sets ?

A set is a collection which does not allow duplicates. Set internally implements equals() method which doesn't allow duplicates.Adding an duplicate element to a set would be ignored .Set interface is implemented in java.util.set package.Set interface does not have any additional methods . It has only collection methods. A set can contain atmost one null value.
ArrayList is an ordered collection.In arraylists order remains same in which they are inserted. But coming to set it is an unordered collection.

```
public interface Set<E> extends Collection<E> {
}
```

Important operations that can be performed on set :

1) Adding an element to set.
2) Removing an element from set.
3) Check if an element exist in set.
4) Iterating through set.

214) Implementations of Set interface ?

1) HashSet
2) Linked HashSet
3) TreeSet

215) Explain HashSet and its features ?

Hashset implements set interface and extends AbstractSet. Features of Hashset are :

1) It does not allow duplicates.
2) It does not gurantee ordering of elements.
3) It is unsorted and unordered set.
4) Performance wise it is recommended to use hashset when compared to other sets because it internally uses hashing mechanism.
5) Allows insertion of nulls.

Note : For efficiency whenever objects are added to HashSet it need to implement the hashCode() method.

```
public class HashSet<E> extends AbstractSet<E>
implements Set<E>, Cloneable, java.io.Serializable
{
}
```

216) Explain Tree Set and its features?

TreeSet implements navigableSet interface and extends Abstract set.It creates collection that uses tree for storage.
Features of Treeset are :

1) It does not allow duplicates.
2) When we retrieve the elements in treeset we will get elements in sorted order.

public class TreeSet<E> **extends** AbstractSet<E>
implements NavigableSet<E>, Cloneable, java.io.Serializable
{

217) When do we use HashSet over TreeSet?

If we want to search for an element in collection and does not want any sorting order we go for HashSet.
82) When do we use TreeSet over HashSet?
TreeSet is preferred
1) if elements are to be maintained in sorting order.
2) Fast insertion and retrieval of elements.

218) What is Linked HashSet and its features?

LinkedHashSet extends HashSet and implements Set interface.
public class LinkedHashSet<E>
extends HashSet<E>
implements Set<E>, Cloneable, java.io.Serializable {
}
Linked HashSet is similar to HashSet but in linked HashSet we maintain order but in HashSet we don't maintain order. Maintaining order means elements will be retrieved in order which they are inserted.

219) Explain about Map interface in java?

A map is an association of key-value pairs. Both keys and values in map are objects.
Features of map :
1) Maps cannot have duplicate keys but can have duplicate value objects.

220) What is linked hashmap and its features?

LinkedHashMap extends HashMap and implements Map.lLinked hashmap gurantees order of elements . Elements are retrieved in same order they are inserted.Linked HashMap uses internally double linked lists to keep insertion order.
The differences between Hashmap and linked hashmap is
1) LinkedHashMap maintains the insertion order while HashMap doesnot maintain order.
2) HashMap if faster for insertion and deletion of elements when compared to linked hashmap. Linked hashmap is preferred only for faster iteration of elements.
public class LinkedHashMap<K,V>
extends HashMap<K,V>
implements Map<K,V>
{
}

221) What is SortedMap interface?

SortedMap extends Map interface.Sorted Map ***maintains sorted order of keys*** in a map.
By default sorted map ***maintains natural ordering*** if we want custom order we can specify using comparator.
public interface SortedMap<K,V> extends Map<K,V> {
}

222) What is Hashtable and explain features of Hashtable?

Hashtable was available before collection framework.
When collection framework was started Hashtable extends Dictionary class and Map interface.
Hashtable offers a convenient way of ***storing key/ value pairs***.
Hashtable ***does not allow nulls either keys or values***.
Hashtable is ***synchronized***.

223) Difference between HashMap and Hashtable?

Difference	HashMap	Hashtable
Synronization	HashMap is ***not synchronized***.	Hashtable is ***synchronized***.
Nulls	HashMap allows atmost ***one null key and any number of null values***.	Hashtable ***does not allow null values***.
Performance	Since HashMap is not synchronized its performance is ***faster than*** Hashtable.	Performance is ***slower*** when compared to HashMap.
Introduction	HashMap introduced starting from	Hashtable is even before collection

	collection framework.	framework.

224) Difference between arraylist and linkedlist?

Difference	Arraylist	LinkedList
Access	Implements RandomAccess interface we can ***search randomly*** all the elements in the list.	It extends Abstract sequential List interface which provides ***sequential access*** to elements.
Searching and retrieval of elements	***Searching and retrieval*** of elements is ***fast*** since arraylist provides random access.	***Searching and retrieval*** of elements is ***slow*** because of sequential access to elements.
Addition and removal of elements	***Adding and removal of elements in random positions is slow***.For example if we want to add element to middle of the list we have to move the elements in the list and then we need to insert the element. Similarly for removing the element we need to follow the same thing.	***Adding and removal of elements in random positions is fast*** because there is no need of resizing the array just by updating the node structures with new addresses.

225) Difference between Comparator and Comparable in java?

Sno	Comparator	Comparable
1.	Defined in ***java.util package***	Defined in ***java.lang package.***
2.	Comparator interface is used when ***we want to compare two different instances***	Comparable is used ***to compare itself with other instance***.
3.	Comparator is used when we want ***custom sorting***.Ex : If we take employee class sorting by employeeId is natural sorting.	Comparable is used for ***natural sorting*** of objects.Ex : If we take employee class sorting by ename and age we can say as custom sorting.
4.	Should override int compare(T o1, T o2) method which takes two instances.	Should override public int compareTo(T o) method which takes one instance.
5.	For sorting objects we use collections.sort(list,new Comparator);	For sorting objects we use collections.sort(list);

226) What is concurrent hashmap and its features ?

Concurrent HashMap is implemented in *java.util.concurrent* package.
Concurrent HashMap extends Abstract Map and implements concurrent Map.
Concurrent HashMap is used in multi threaded environment.
]It is similar to Hashtable and synchronized version of hashmap but with minor differences.
Concurrent HashMap *does not allow null keys and values*.

227) Difference between Concurrent HashMap and Hashtable and collections.synchronizedHashMap?

Locking Mechansim :ConcurrentHashMap uses completely different hashing mechanism called *lock striping* which offers better concurrency and scalability.
The main advantage of this mechanism is better concurrency instead of synchronizing every method by using common lock which allows only one thread to access at a time, *it allows better concurrency by allowing multiple threads to access*.
ConcurrentModificationException :ConcurrentHashMap provides iterators which doesnot throw concurrent modification exception which allows only one thread to access iterator, *while synchronized map may throw concurrent modification exception*.

228) Explain copyOnWriteArrayList and when do we use copyOnWriteArrayList?

copyOnWriteArrayList is used in multithreaded environment. If we want to iterate over arraylist ,but the arraylist is updated by other threads to prevent concurrent modification exception we have two solutions :
1) First one is we need to synchronize the list manually by using collections.synchronized(list) and iterate over the list in synchronized block to avoid concurrent modification exception.
2) The second one is to use copyOnWriteArrayList which takes care of concurrency.

The advantage of using copyOnWriteArrayList is no need to synchronize list explicitly. So when we use copyOnWriteArrayList when a thread modifies the list while the other thread was iterating it does not modify original list but creates a copy of list with modified contents so that the iterator won't know the modifications made to original list.

229) Explain about fail fast iterators in java?

When iterator iterates over collection, collection should not be modified except by that iterator. Modification means collection cannot be modified by thread when other thread is iterating, if such modification happens a concurrent modification exception will be thrown.Such kind of iterators are fail fast iterators.
Ex : ArrayList,HashSet,HashMap. Almost all the iterators implemented in collections framework are fail fast.

230) Explain about fail safe iterators in java?

Fail safe iterators are iterators which does not throw concurrent modification exception, when one thread modifies collection and other thread in the process of iterating the collection.
It does not throw concurrent modification exception because when other thread was iterating it does not modify original list but creates a copy of list with modified contents so that the iterator won't know the modifications made to original list.
Ex : copyOnWriteArrayList

Core java Serialization interview questions

231) What is serialization in java?

Serialization is the process of converting an object in to bytes, so that it can be transmitted over the network,or stored in a flat file and can be recreated later. Serialized object is an object represented as sequence of bytes that includes objects data, object type, and the types of data stored in the object.

232) What is the main purpose of serialization in java?

The main uses of serialization are :
1) Persistence:
We can write data to a file or database and can be used later by deserializing it.
2) Communication :
To pass an object over network by making remote procedure call.
3) Copying :
We can create duplicates of original object by using byte array.
4) To distribute objects across different JVMs.

233) What are alternatives to java serialization?

XML based data transfer
JSON based data transfer.
XML based data transfer : We can use JIBX or JAXB where we can marshall our object's data to xml and transfer data and then unmarshall and convert to object.
JSON based transfer : We can use json to transfer data.

234) Explain about serializable interface in java?

To implement serialization in java there is an interface defined in java.io package called serializable interface. Java.io.Serializable interface is an marker interface which doesnot contain any any methods. A class implements Serializable lets the JVM know that the instances of the class can be serialized.
Syntax:

```
public interface Serializable {
}
```

235) How to make object serializable in java?

1) Our class must implement serializable interface.If our object contains other objects those class must also implement serializable interface.
2) We use ObjectOutputStream which extends OutputStream used to write objects to a stream.
3) We use ObjectInputStream which extends InputStream used to read objects from stream

236) What is serial version UID and its importance in java?

Serial version unique identifier is a 64 bit long value .This 64 bit long value is a hash code of the class name,super interfaces and member. Suid is a unique id no two classes will have same suid. Whenever an object is serialized suid value will also serialize with it.
When an object is read using ObjectInputStream, the suid is also read. If the loaded class suid does not match with suid read from object stream, readObject throws an InvalidClassException.

237) What happens if we don't define serial version UID ?

If we don't define serial version UID JVM will create one suid for us. But it is recommended to have suid rather than JVM creating because at run time JVM has to compute the hashcode of all the properties of class. This process makes serialization low. We can't serialize static fields one exception to this is suid where suid gets serialized along with the object.
Ex :private static final long serialVersionUID = -5885568094444284875L;

238) Can we serialize static variables in java?

We can't serialize static variables in java. The reason being static variable are class variables that belongs to a class not to object, but serialization mechanism saves only the object state not the class state.

239) When we serialize an object does the serialization mechanism saves its references too?

When we serialize an object even the object it refers must implement serializable then the reference objects also get serialized. If we don't make reference objects serializable then we get NotSerializableException.

240) If we don't want some of the fields not to serialize How to do that?

If we don't want to serialize some fields during serialization we declare those variables as transient. During deserialization transient variables are initialized with default values for primitives and null for object references.