INTERVIEW CLASS

1. Abstract.
2. Interface.
3. Method Referencing .
4. Constructor
5. Reflect
6. Collection

1. Abstract

Q1 Ky hum abstract or final (keyword) ko ek sath use ker sakte hai ?

Ans:- No , we can not use final and abstract keyword together .Because both are just opposite.

Reason:- abstract class ko inherit karna compulsory hota hai and hum final ko inherit nhi kar sakte hai. Yadi kiya to error dega illegal combination of modifier abstract and final.

Q2:- ky hum abstract ke sath static ya private ka use ker sakte hai?

Ans:- No, we can not use static ya private with abstract keyword together .Because both are just opposite.

Reason :- abstract run time polymorphism hota hai and static compile time polymorphism hota hai.

Q3 Type of Methods in java?

Ans:- There are two type of method in java:-

1. **Predefine method.**
2. **User-define method.**

Predefine method:- the methods which are provided by java.

User-define method :- this are two type :- i. static (ii) instance

Instance method can be abstract , final , private.

Q4 Bina Body vali method ko ky bolte hai?

Ans:- abstract method .

Q5 Body vali method ko ky bolte hai?

Ans:- Concrit method.

Q6 What is abstract class in java ? Explain.

Ans:- * Abstract class is declare using abstract keyword .

- **Abstract class can contain abstract and non-abstract both type of methods.**
- **We can not create object of abstract class.**
- **To access the data of abstract class it is compulsory to inherit the abstract class.**

Q7 What is abstract method in java? Explain.

Ans:- A method without body is known as abstract method.

- It is declared using abstract keyword.
- If class contains any abstract method. So it is compulsory to declare that class as abstract class.
- It is compulsory to inherit the abstract class if we want to access the data of abstract class.
- It is compulsory to define the body of abstract methods in sub class(means Override). Otherwise we need to make subclass also abstract class.

Q8 What is the purpose and advantage of making abstract method?

Ans:- yadi hum same method se different – different operation perform karvana cahate hai or vo operations Object pr depend honge ess purpose se hum abstract method bana ker ready karte hai.

- Same method hogi operations different honge.

Q9 What is Concrit class in java?

Ans:- A class which does not have any abstract method. That class is known as concrit class.

- Jitni bhi java me normal class hai vo sabhi concrit class kahlati hai.
- Concrit class ko complete class bhi kahte hai. Kyo ki eske ander jitni bhi methods hoti hai unka complete implementation hota hai.
- Hum Concrit class ka object bana sakte hai but abstract class ka nhi bana sakte hai

Q10 Difference between abstract class and concrit class?

Ans:-

Abstract class	Concrit class
• abstract class is declared by abstract Keyword.	It is not declared by abstract keyword.
• It is also called incomplete class.	It is called complete class.
• It can have abstract and concrit both Type of method.	It can have only concrit method not abstract.
• Yadi hume kisi class ka partial Implementation karna hai to abstract class ka use karege.	hume yadi kisi class ka complete implementation karna hai to concrit class ka use karege.
• Hum abstract class ka object create nhi ker sakte.	eska ker sakte hai.
• abstract class ko kabhi bhi final nhi Bana sakte.	Esko bana sakte hai.

Q11 Ky hum userdefine JRE bana sakte hai?

Ans:- yes by JLink (java Linker).

Q12 What is JLink?

Ans:- JLink is used to create our own customized small JRE(Java Runtime Environment). JLink is the java's new command line tool (available in JDK_Home/bin) which allows us to link set of only required modules(and there dependencies to create runtime environment).

Q13 Hume java ne JLink ki facility kab se provide ki hai?

Ans:- version 1.9 se.

Q14 Difference between abstract class and interface ?

Ans:-

abstract class	Interface
• Abstract class can not refer lambda Expression.	Functional interface with default method can refer Lambda expression.
• Inside abstract class we can override Object class method.	Inside interface we can not override object class Method.
• Abstract class can have abstract and And non-abstract method.	Interface can have only abstract method. Since java 1.8 it can have default and static method also.Since 1.9 private also.
• Abstract class does not suppor Multiple inheritance.	Interface supports multiple inheritance.
• Abstract class can have final, non-Final ,static and non-static variable.	interface has only final or static variable.
• The abstract keyword is used to Declare abstract class.	The interface keyword is used to declare interface
• Abstract class can provide the Implementation of abstract class.	Interface can not provide the implementation of interface.
• An abstract class can extends another Java class and implements multiple Java interface.	An interface can extends another java interface only.
• An abstract class can be extends using extends keyword.	An interface can be implemented using keyword Implements.
• A java abstract class can have class member like private , protected etc.	An interface can be implemented using keyword implements member of java interface are public by default.

Q What is marker interface ?

Ans:- Esa interface jisme koi bhi method na ho usse hi marker interface kahte hai.

Q What is functional interface?

Ans:- Jis interface me single abstract method ho usse hi functional interface kahte.

Q Functional interface ka use ky hai?

Ans:- Lambda expression ke purpose se functional interface ka use karte hai.

Ex:- actionListener -> abstract single method.

Q What is method referencing ?

Ans:- Predefined method ki body ko access karne ke purpose se method reference ka use karte hai.

Q What is multi threading?

Ans:- Ek hi software me multiple process work kr rhi ho usse multi threading kahte hai.

Q Interface me default method ki facility kb provide hoi hai?

Ans:- java version 1.8

Q Interface ki functionality ka java ne kha use kiya hai?

Ans:- JDBC (API hi kahte hai)

How:- java ne interface (means abstract method) API ke form bna di hai and sab database ko java se connectivity karni hai (to ye sab apni -apni tarha se method bnayege to java confuse ho jayega). Meanse interface bna diya hai(API ke form me) jisme java ne apni method (abstract method)bna di hai , so (all database :- Sql , Oracle, mariyo etc)apni coding java ke banye haye interface ke hisab se karege to vo coding java ko bhi samaj me aa jayegi and one interface is work on all. So , it is usefull (JDBC connectivity).

Q Interface me static method ki functionality kb provide hoi hai ?

Ans:- java version 1.8

Q Interface me private method ki functionality kb se provide ki hai?

Ans:- java version 1.9

Q Why use java interface ?

Ans:- It is used to achive abstraction.

- By interface we can support the functionality of multiple inheritance.
- It can be used to achive loose coupling.

Q Relationship between classes and interface ?

Ans:- * class class ko extends karti hai.

- Class Interface ko implements karti hai.
- Interface interface ko extends karta hai.
- Interface ki method by default public hoti hai to subclass me public likhna compulsory hai.

Q Difference between concrit and abstract method ?

Ans:-

Concrit method	abstract
• Concrit method me fully implementation Hoti hai.	Abstract method me partially implementation hota hai.
• Concrit method ke sath abstract keyword Ka use ker sakte hai.	Abstract method ko abstract keyword se declare karte hai.
• Concrit method ke sath private,final,static Keyword ka use ker sakte hai.	esme nhi ker sakte hai.

• Concrit method me esa koi rule nhi hoti Hai.	Abstract methd jis class ke ander hota Hai , us class ko bhi abstract declare karna Compulsory hota hai.
• Body wali method ko concrit method Bolte hai.	Bina body wali method ko abstract method bolte hai.
• Concrit method can be private.	Abstract method can not be private.

Q Abstract class me constructor bana sakte ya nhi ?

Ans:- ha bana sakte hai.

- Sub class ko object se super class ka constructor call ho jata hai.

Q What is the advantage of calling super class constructor using subclass Object ?

Ans:- Esse ye benefit hai ki yadi suppose humare pass 50 classes hai or usme kuch same property hai to har class me define karna hoga eske vajaye hum kyo na direct ek class bana de jo ki super hogi and sabhi child class se super keyword ka use ker ke parameter pass ker de to vo automatic call ho jayega.

2. Interface

Q What is interface ?

Ans:- interface is a keyword.

- Upto java version 1.7 we can declare only abstract method inside the interface .But from java version 1.8 we can create default method with body using default keyword. We can also create static method.
- Yadi hum default method ko default keyword se bhi declare karte hai tab bhi internally usse as public hi treate karta hai esliye hume default method ko bhi override karte time public access spacifier hi use karna padta hai.
- From java version 1.9 we can create private method also inside interface.
- All the method inside interface are bydefault public and abstract.
- All the fields(variable) inside the interface are by default public static and final.
- We can not create instance(Object) of interface.
- If we want to access interface data then we need to implements the interface in a class & implementation class Object can access interface data.
- To implements interface in a class we us implements keyword.
- A class that implements interface must implement all the abstract method declare in the interface.

Q Can we override static method in interface ?

Ans:- You can not override static method of interface . You can just access them using the name of interface . If we try to override the static method of an interface by defining a similar method in the implementation class. It will consider as another static method of the class.

- Jo humne another static method banai hai usse another static method consider karega error nhi dega.
- We can access interface static method only using the interface name not using Object.

Q What is default method ? Use of default method.

Ans :- A method which is declare using default keyword inside an interface is called default method. It is use inside interface from java verion 1.8 onward.

USE :- Before java version 1.8 when we didn't have concept of default method . So that we could create only abstract method inside interface. In this case suppose 1000 classes implemented that interface the if we want to add any new method in interface so we had to override that method in all 1000 classes. Due to that reason OR come out from that problem . They gave us default method concept from java version 1.8 now if we want to add any method inside interface so we can create.Then now we don't need to override that method inside all the classes. If we want to use that default method so we can call that by using class Object.

- They gave us one more facility that if we want to change that default method definition so we can change inside our class by using overriding concept.

Q What is reference variable?

Ans:- A reference variable is like a container.

- A reference variable point to a memory block or an object.
- Reference variable we use for multiple use

Q Is it possible to have a abstract method inside the final class?

Ans:- No abstract method can't be declare inside the final class.

- Abstract :- incomplete method.
- Final :- complete class.

Q Is it possible to have a final method in a abstract class?

Ans:- Yes.

Q Is it possible to have a static method in a abstract class ?

Ans:- Yes.

Q Jis class ke ander main method hai us class me abstract method bna skate hai ya nhi ?

Ans:- Yes.

Q Is it possible to have a private method inside the abstract class?

Ans:- Yes.

Q Is it possible to have a static final method inside the abstract class ?

Ans:- Yes bna sakte hai.

Q What is the difference between abstract keyword and final keyword?

Ans:-

Abstract Keyword	final Keyword

• abstract keyword ka use partial Incomplete implementation ke purpose Se karte hai.	for complete implementation .
• Inheritance possible (both class and method)	Method ko inherit ker sakte hai but class ko nhi .
• abstract class ka object nhi bana sakte hai.	Final class ka bna sakte hai.
• Abstract keyword ko hum method or class Ke sath use ker sakte hai.	final keyword ko hum method, class or Variable ke sath use ker sakte hai.

Q What is the difference between access specifier and access modifier(non- access specifier)?

Ans:-

Access specifier	Access modifier
• Keyword:- default,public,private, protected.	Keyword:- final, static, abstract ,synchronized.
• Scope and access karne ki functionality Provide karne ke purpose se access Specifier ka use karte hai.	class method and variable ko kuch extra functionality provide karne ke purpose se non-access specifier ko use karte hai.

- Implemetation class me bina effect show hoye interface me changes apply karne ka bhi ek advantage hai jab hum default method ka use karte hai.

Q Can we have default method definition in the interface without spacified the keyword default?

Ans:- no error dega "interface abstract method can not have body".

Q Can an interface contain more than one default method?

Ans:- Yes we can create N number of default method in an interface.

Synchronized :- It has the capability to control the access of multiple threads to any shared resource(thread saf -> procceslow).

Q Why we can not create the object of abstract class ?

Ans:- Because abstract class is incomplete class(abstract class incomplete / restraction based class hoti hai).

Q Hum abstract method kab bana kr ready karte hai?

Ans:- Runtime polymorephism achive karne ke purpose se hum abstract method bana ker ready karte hai.

Q Difference between reference variable and Object?

Ans:-

Reference variable	Object
• Future me bar-bar use karne ke Purpose se reference variable ka use Karte hai.	Object me hume class ke structure tak hi use karne ki permission hai.

• Reference me class load nhi hoti hai.	Object me class load hoti hai.
• A reference is just a variable that points To the location of the object memory.	An Object contains element such as methods and variable such as state and Behaviour.

Q .class file kab load hoti hai?

Ans:- .class file runtime pr load hoti hai.

- Class loader loads the .class file the java class loader is a best of the java runtime.

Q Which access specifier and modifier can be used with abstract class?

Ans:- Valid:- public , protected , and default.

InValid:- private, static ,final.

Q Abstract class ke ander abstract method ka hona compulsory hai ya nhi ?

Ans:- Nhi, dono ho sakti hai abstract and non-abstract .

Q Final class ke ander final method ka hona compulsory hai ya nhi ?

Ans:- Nhi.

Q Abstract class ke ander body wali normal method hai to vo abstract class ke ander se access kr sakte hai ya nhi ?

Ans:- No, Because it is abstract class and abstract class ka object nhi bana sakte hai.

Q super class ka reference sub-class ke object ko hold kar sakta hai(benefit)?

Ans:- Runtime pr bhi object create ker sakte hai.

- Super class ka reference variable har type ki sub-class ka object store ker sakta hai.
- Like, command line argument par directly name of class(mean) – student , employee , pass karna ho to vo super class ke pass jaker hold kar sakte hai.

Q Bina body wali method banana ke fayde ?

Ans:- ATM.

Q Can we use more than one default method with same name in different interface and implements by same class of both interface by coma(,) seprate.

Ans:- No , But if sub-class interface will extends super class interface then class will implements only sub-class interface so it will call sub-class method.

Q What is the use and advantage of private method inside the interface?

Ans:- from java 1.9 version private methods can be added to interface in java.

- private method can be implemented by static or non-static This means that in an interface we are able to create private method to encapsulation code from both default and static public method signature within the interface other statically defined method can make use of these private method.

- Interface ke variable ko interface ke name se access karte hai.

  Default method ki body dena compulsory hai interface me otherwise "missing body or declare abstract " error ayegi. yadi humne abstract declare ker diya tab bhi error dega "illegal combination of modifier : abstract and default."
- Hum default abstract method ki body nhi bana sakte hai error ayeggi "illegal combination of modifier : abstract and default."
- Hum static default ko bhi sath me use karke method nhi bana sakte hai.

Q When we use extends and implements ?

Ans:- * interface extends interface.

- Interface extends more than one interface.
- Class extends class .
- Class implements interface.
- Class implements more than one interface.

Q What is SAM?

Ans:- SAM (Single Abstract Method) interface eska hi dusra name functional interface kahte hai.

- It has only one abstract method.
- Version 1.7 feature String use in switch() bracket , main method compulsory ,dimond operator.

Annonymous Inner class.

```
Ex:- class Demo{
psvm(String arr[]){
Inter1 i=new Inter1(){
        Public void show(){
        S.o.pln("Ram");
        }
        };
        i.show();
}
}
```

Lembda Expression:-

```
Ex:- interface Inter1{
void show();
```

```
}
class Demo20{
        public static void main(ar[]){
        Inter1 i=()->{
        S.o.pln("ram");
        };
        i.show();
        }
        }
```

- **Templet class ka advantage:-**
- ClassCastException and typecasting se bach sakte hai.
- Predicate ek functional interface hai. Jo ki import java.util.function.*; package me rakha hai and test() uski abstract method hai.
- Predicate interface version 1.8 me hi intraduce hoaa hai kyo ki functional interface ki facility version 1.8 se hi aai hai.
- Annotation ye batata hai ki jo hum karna cahate hai vo successfully hoaa hai ya nhi Ex:- for overriding **@Override** ,for Check Functional interface **@FunctionalInterface** annotation ka use karte hai.
- Annotations humesha Capital latter se hi start hoti hai.
- Functional interface ke ander only 1 hi abstract method ho sakti hai but kitni bhi default, static and private method ho sakti hai.
- Predicate interface me 1 abstract method 2 static method and 3 default method hai.
  abstract method:- return type= Boolean
  i. test()
  static method :- Return type =Predicate
  i. isEqual()
  ii. not()
  default method :- Return type =Predicate
  i. and()
  ii. or()
  iii. negate()
- Predicate interface ki methods ka return type Predicate esliye hai kyo ki ye indicate kar raha hai ki eski methods method chaining ko support kerti hai.
- **and()** method me dono condition true hona cahiye . **or()** method me ek bhi condition true hogi to work karega. **negate()** method me jo condition satisfied hogi usko chhod ker sabhi element dega.

Q Why we use functional interface with lambda expression?

Ans:- kyo ki lambda expression me hum method ka name nhi dete hai mtlab usme ek hi abstract method hona cahiye . Jo ki only functional interface me possible hai esliye hum lambda expression ke sath functional interface ka use karte hai.

Q Predicate ka use ker ke prime number ka program.

Q Predicate ka use ker ke palindrome number ka program.

Q Predicate ka use ker ke prime and palindrome both number ka program.

Q Predicate ka use ker ke jo string a se end ho rahi hai uska program.

Function interface:- Function interface bhi ek functional interface hai.

- **Predicate interface** ke pass jese **test()** name ki abstract method thi vese hi **Function interface** ke pass **apply()** name ki abstract method hai.
- **test()** method me hum kisi bhi type ka data pass ker sakte hai but uska return type boolen hi hota hai. Jabki **apply()** method me hum kisi bhi type ka perameter pass ker sakte hai and uska return type bhi kuch bhi de sakte hai.
- Function interface ke pass total 4 method hai. 1 abstract 2 default and 1 static.
- **Abstract :-**apply() **,default:-**i. compose() ii. andThen() **, static :-** identify()

Q Difference between compose() and andThen() method?

Ans:- * compose() always solve right to left andThen() always solve left to right.

- compose() always solve before andThen() andThen() always solve after compose().
- Lambda expression me vo automatically reciver jis type ka hai vo automatically uski method ko override ker lega.
- Jo left hand side me rakha hai vo automatically right hand side work karega due to type inference.
- Yadi direct return a->a ker diya to vo fer return type me jo interface rakha hai uski method ko override ker lega but vo functional interface hona cahiye otherwise error dega.

Q What is type eraser ?

Ans:- To implement generic behaviour, java compiler apply type erasure. Type erasure is a process in which compiler replaces a generic parameter with actual class or bridge method. In type erasure, compiler ensures that no extra classes are created and there is no runtime overhead.

Consumer Interface

- Ye bhi ek functional interface hai .
- Eske pass 1 abstract method hai accept() Eska return type void hai. Kyo kiye only templet type ka Parameter leta hai.
- Eske pass 1 hi default method hai **andThan()**
- Eske pass koi bhi static method nhi hai.
- For check all methods of class Use the command javap java.packageName.ClassName

Supplier Interface

- Ye only supply kerne ka kam karta hai kuch consume nhi karta hai.esliye eska return type templet hota hai and koi perameter nhi leta hai.
- Eske pass only one hi abstract method hai .Jiska return type templet hai. **Ex :- T get();**

All 4 interface abstract methods Name:-

Predicate<T>

Boolean test(T t);

Function<T,R>

R apply(T t)

Consumer<T>

void accept(T t);

Supplier<T>

T get();

- Hum Array type ka bhi generic type de sakte hai. Ex:- Inter1<String[]>i=(a)->a; Kyo ki array bhi as a Object ki tarha hi treat hota hai.
- 1 se jiyada perameter ke purpose se **BiPredicate , BiFunction , BiConsumer** interface provide kiye hai.

BiPredicate :-

- BiPredicate ke pass koi bhi abstract method nhi hai ye **test()** ka use karta hai.
- 3 default method hai **and() , or() , negate()**. But koi bhi static method nhi hai.
- BiPredicate me hum only 2 hi perameter de sakte hai na hi 2 se jiyada na hi 2 se kam.

BiFunction :- eske pass bhi koi abstract method nhi hai ye **apply()** method ka hi use karta hai.

- Only 1 default method hai **andThen()**

BiConsumer:- ye bhi **accept()** abstract method ko hi use karta hai. And only 1 abstract method hi hai andThen().

- Only Predicate , Function , Consumer interface ke corresponding hi BiPredicate , BiFunction , BiConsumer interface provide kiye hai . Supplier ke corresponding nhi.

- Humesa Operation datatype ke corresponding hi perform hota hai. Esliye jab bhi hum Generic type dete hai to vo pahle corresponded Primitive data type me convert hota hai then operation perform hota hai.

- Esliye hi hume Primitive type functional interface bhi provide kiye hai. Taki hum boxing unboxing se bach sake.

Predicate Primitives interfaces :-

- IntPredicate
- LongPredicate
- DoublePredicate

Function Primitives interfaces :-

- IntFunction
- LongFunction
- DoubleFunction

Consumer Primitives interfaces :-

- IntConsumer
- LongConsumer
- DoubleConsumer

Supplier Primitives interfaces :-

- IntSupplier
- LongSupplier
- DoubleSupplier
- BooleanSupplier

Two Extra interfaces

- ActionListener
- ItemListener

3 Method Referencing .

- Hum interface ki abstract method ko kai tarha se access/Override ker sakte hai **Like By anonymous class , By Lembda Expression , By method Referencing.**

Q What is Method Referencing ?

Ans:- Pahle se bani hoi method ko use karna by(::) method Referencing operator is called method referencing . That method can be user define or predefine. We can say it Is advance version of lambda expression.

- **There are three types of method reference :-**
  - i. Reference to static method
    Ex:- class_Name :: method_Name;
  - ii. Reference to instance method
    Ex:- Class_Name obj=new Class_Name();
    obj::Method_Name;
  - iii. Reference by constructor
    Ex:- class_name :: new;
- **Method reference concept is used in stream API.**
- Parameter list same hona cahiye .
- Return type se koi fark nhi padta always void return karta hai. Kyoki yadi hum System.out.print(i.show()); karte hai to error deta hai void type not allowed here.
- Ye bhi functional interface ke sath hi work karega kyo ki ye lambda expression ka advance version hai.

4 Constructor

Q what is Constructor ?

Ans:- Constructor is a special member function whose name is same as class name.

- it does not have any return type.
- When we create an object constructor is called automatically.
- It is use to initialize the instance variable. It is also use for initialize the object.

Q what are type of constructor?

Ans:- there are three type of constructor :-

i. Default ii. Non-parameterized iii. Parameterized .

i. **Default constructor :-** It is created by compiler if not any explicit constructor is created . It does not contains any parameters.
ii. **Non-parameterized :-** A constructor which is created by programmer without any parameter is called Non-parameterized constructor.
iii. **Parameterized Constructor :-** A constructor which is created by programmer with parameter is called parameterized constructor.

Q Ese konse predefine class hai jiska object create nhi ker sakte hai ?

Ans:- Math class and Runtime class jo ki GC ke accoding ati hai.

Q What is use of private Constructor ?

Ans:- If we want to apply restriction on object creation then it can be achieve by making the constructor private. If we does not want to create object externally and we want that object can be created internally then we make constructor private.

Q Whenever we are creating child class object parent class object will be created or not?

Ans:- Whenever we are creating child class object parent class object will not be created. When we call constructor using super().In java for every object unique hashCode is generated . So if we want to check this so we can call hashCode() method in child and super both class so both class hashCode will be same that is prove that parent class object does not created. Because for every object different hashCode is generated.

Q ky bina super class ka object create kiya super class ka constructor call or not?

Ans:- yes with using super keyword.

Q what are the use of parameterized constructor?

Ans:- the parameterized constructor is used to initlize the instance variable with value.

Q What is Constructor chaining ?

Ans:-Child class ke object se parent class ke constructor ko call karna hi constructor chaining kahlata hai.

OR

Calling one constructor from the another constructor is called constructor chaining.

Q What is constructor invocation in java?

Ans:- When we call one constructor by another constructor and again when we call that constructor by that called constructor so that error is come. It is like recursion but in this we call call constructor recursively.

Q what are the difference between Error and Exception ?

Ans:-

Error	Exception
• All the Errors in java are unchecked.	It can be Checked or Unchecked.
• It can not be recovered	It is recoverable.
• It belongs to java.lang.Error package.	It belongs to java.lang.Exception package.
• It is not know to compiler.	It is known to compiler.
• It is mainly coused by environment in Which application is running.	It is caused by application by itself.
• Ex:- StackOverflow , VirtualMachin Error, OutOfMemoryError.	SQLException , IOException , ArrayIndexOutOfVoundsException.
• It occur at run time only.	It occur at compile and runtime.

Q What is advance of lambda expression?

Ans:- Fewer line of Code:- It reduce the line of code. It can only used with functional interface.

- Higher Efficiency :- By using Stream API and lambda expression, we can achieve higher efficiency (Parallel execution) in case of bulk operation on collection.
- Code is directly readable without interpretation.
- We don't need to know much information about method or interface.

Q What is advantage of method referencing?

Ans:- You can use it to write clean code in java .The main advantage is that. They make the code even shorter by eliminating lambda expression , Which make code more readble.

Q What is constructor chaining ?

Ans:-Constructor chaining is the process of calling one constructor from the another.

- Constructor with respect to current object.
- Constructor chaining can be done by two ways:-

i. **With in same class:- It** can be done using this() keyword for constructor in same class.

ii. **From base class:-** by using super() keyword to call constructor from the base class.

Note:- Constructor chaining occur through inheritance. A sub class constructor's task to call super class's constructor first. This ensures that creation of sub class's object starts with the. Initialization of the data member of super class. There could be any number of classes in the inheritance. Every constructor calls up the chain till class at the top is reached.

- Java ke hisab se Arraylist ek Ex:- hai. Constructor chaining.

Q What is advantage of instance block?

Ans:- The main advantage of instance block is instance block are executed wherever an object of any kind is created. If we want to execute on the creation of all kinds of objects, then using instance block is a good idea to avoid writing the same logic inside every constructor.

Q What is difference between private, default, protected , public .

Ans:- **private**:- private can be access in the same class only.

default:- default can be access in the difference class but in the same folder/package. It is not access in difference folder.

protected:- It can be access in the same folder or another folder also but there should be a inheritance applied in the both classes.

- Default data different folder me access nhi ho sakta hai.
- Protected same folder me access ho jayega but different folder me access Karne ke liye inheritance apply karna hoga.
- Jo compiler constructor create karta hai default vala uska access spacefier same class ke access spacifier jesa hota hai.
- Jab hum bina parameter vala manually banayege to vo by default default hi hoga

Q What is access specifier of default constructor?

Ans:- It will be same as class access specifier would be.

- Class ke case me hume manually access spacifer likhna hota hai method or variable ka but method ke case me jesa access spacifer method ka hoga vhi variable ka bhi hoga.

Q What is difference between Anonymous inner class and Lambda expression?

Ans:-

Anonymous inner class(AIC)	Lambda expression
• It is class without any name.	It is method without name(Anonymous Function).
• It can implements an interface that contain any number of abstract methods	It can only implements an interface which contains single abstract method.
• It can extends abstract and concrit Class	It can not.
• Inside AIC we can declare instance Variables.	It does not allow to declaration of instance variable Whether the variable declare simply act as local Variable.
• AIC can be instantiated.	It can not be instantiated.
• It is good choice if we want to handle multiple methods. At the time of compilation a separate .class will be generated.	It is good choice if we want to handle interface single method. At the time of compilation a separate .class will Not generated.It simply convert into private method of outer class.

- Java me protected data ko different folder me access karne ke liye inheritance apply karna compulsory hai.

- Protected data sub class me access to hota hai but Object create ker ken hi ker sakte super keyword ka use karna hoga.
- Protected ke sath static use ker sakte hai.
- Protected constructor ko access karne ke liye constructor chaining ka use karna jaruri hai eske bina protected constructor access hi nhi hoga.
- Ek hi package me or ek hi program me 1 se jiyada pubic class nhi ho sakti hai.

Q can we have super() and this() in the same constructor?

Ans:- No we can not have both **super()** and **this()** in the same constructor. Because **super()** and **this()** must be first statement in the constructor. So we can not use both in same constructor.

Q What is the difference between super() and this()?

Ans:- **super() :-** it is use to call super class constructor.

this() :- It is use to call same class constructor.

Q Can we use this() and super() in a method?

Ans:- We can use super() and this() only in constructor not anywhere else any attempt to do so will lead to compile time error. Both this and super are non static and can not be used in static context. Which means you can not use this and super keyword inside main method failing to do so will result compile time error. **"non- static variable this can not be referenced from static context".** Same in true for using super keyword inside main method.

Q What is difference constructor?

Ans:- When we do not create any constructor in class then at compile time java compiler by default supply a no argument constructor which is called as default constructor which is called as default constructor in java.

Q What is a no-arg constructor?

Ans:- A constructor which does not contain any argument is called as no argument constructor.

- Yadi different package me hai to protected constructor ko hum inheritance se hi only child class me super keyword se hi access ker sakte hai. Object se nhi ker sakte.
- Bina kuch change kiye class A me bina inheritance ka use kiye class A ka static block access karna to **Class.forname("className")** main method me likhne se ho jayega kyo class load ho jati hai .es case me object create nhi hota only class load hoti hai eska prove ye hai ki constructor call nhi hota hai.only static block call hota hai.

Q Can we have class without any constructor? What will happen during object creation?

Ans:- Yes we can have a class without any explicit constructor in it. When we compile the program then compiler will automatically add no argument constructor which is called default constructor. It's access modifier will be same as class will have.

Q Can a constructor return any value ?

Ans:- yes constructor return class object.

Q Can we create Object without new keyword?

Ans:- Yes we can create Object without new keyword by using predefined method of **Class** class method **newInstance()** . Ye 2 exception throws karti hai **InstantiationException** and **IllegalAccessException**. Esme RunTime pr decide hota hai kis class ka Object create karna hai.

- Class.forName("className") ka return type Class hi hota hai.
- newInstance() method ka return type Object hai.

Q What is difference between import java.util.*; and import java.util.ClassName;? Means what is difference between * and direct className?

Ans:-ClassName se import karne se program ki readability badh jati hai. And Sabse main ClassName se import karne pr .class file junrate hoti hai. * se import karne pr nhi hoti hai .class file junrate.

- Yadi * se import kiya to us package ki sari classes ko access karne ki permission mil jati hai.
- Yadi * se import kiya to us package me same name se class hone pr conflict ho jayega.
- newInstance() method 0 parameter nhi leti hai.

Q When instantiationException is come?

Ans:- when we create parameterized constructor and create Object by using newInstance() method.

Q What is IllegalException?

Ans:-When we create object of private constructor using newInstance() method.

- When we create object of interface using newInstance() method.
- When we create object of abstract class using newInstance() method.

Q What is dynamic type language? And why java is called dynamic type language?

Ans:- Due to var keyword . Automatically type ka anuman laga lena hi dynamic type language kahlati hai.

Q Why java is dynamic language ?

Ans:- class loader ke karan kyoki java tab tak class ko load nhi karega jab tak ke requirement nhi hogi.

- Yadi other package se class ko access karna hai to Class.forName("Path") me pura path dena hota hai means pura package name with class name.
- class ke name se import karne pr .class file junrate hoti hai.**Like import java.demo1.io1.A;**
- But yadi * se import kiya to .class file nhi junrate hoti hai.**Like import java.demo1.io1.*;**
- Kyo ki esme compile time pr pta hi nhi ker pata hai ki kis class ke coursponding working karna hai. Kyo ki .class file to compile time pr hi junrate hoti hai.

Q Can a constructor return any value ?

Ans:- No constructor does not have return type but implicitly it returns instance of class.

Q What happens when constructor is defined as protected?

Ans:- In genrel protected method can be accessed by other class in a different package only through inheritance. But when you assign protected a constructor it behaves a bit different. By a call of super() (according to JSL(Java Library Spacification))

- private constructor is accessible in its own class where defined. A class with private constructor can not be inheritance by other class used in singleton design pattern used for extra security provision not to allow the instantiation in other class.
- default constructor is accessible within the same class and also for all the classes within the same package.
- A class with both private and protected constructor available can be inherited. But subclass has the accessibility for protected constructor only.

Q Why we can not use final keyword with constructor?

Ans:- We use final keyword with method so that it can not be overridden in subclass. But constructor can not be inherited so it can not be overrident also. So there is no need to use final keyword with constructor. That's why we can not use final keyword with constructor.

Q Can we create abstract constructor ?

Ans:- No an abstract method is a method without body so we need to override the abstract method in subclass but constructor can not be inherited so it can not be overridden that's why we can not make constructor abstract.

Q Why constructor can not be declared as static in java?

Ans:- If a method is static it refers to the class but constructor can not be refer to a class it refers to the object that's why we can not declare constructor as static in java.

Q How compiler and JVM can differentiate constructor and method definition of both having same class name?

Ans:- If it contain return type then compiler interpret it as method. If it does not contain return type then compiler interpret it as constructor.

Q In which situation it is mandatory for the developer to provide constructor explicitly ?

Ans:- If we want to initialize the instance variable with a particular value then we need to provide a constructor explicitly.

Q What is the use of constructor in java?

Ans:- It is used to initialize the instance variables.

OR

It is used to initialize the Object.

Q Does Constructor create the object?

Ans:- No, constructor does not create the object, it is called an object is created.

Q What are the rules for defining a constructor ?

Ans:- The constructor must be same as class name.

- It does not contains any return type.

- It can not be static , final or abstract.

Q Difference between constructor and method in java?

Ans:- Name:- constructor have same name as class name. But method can be same as class name or may be other.

Return type:- constructor does not have any return type. But method contains return type.

- Constructor are called automatically when we create object. But we have to call method explicitly.
- Constructors are called in particular order. There is no such facility for method.

Q What is difference between Constructor and Method?

Ans:-

Constructor	Method
• The purpose of Constructor is to initialize the Object of a class.	The purpose of Method is to perform a task.
• Constructor can not be static ,final ,abstract and synchronized.	Method can be static ,final ,abstract and synchronized.

Q When does compiler provide with default constructor ?

Ans:- A compiler provide a default constructor when there is no any other constructor in class.

Q What is the access modifier of the default Constructor ?

Ans:- It is same as class access modifier.

Q What are the access modifier that can not be applied to a constructor?

Ans:- Constructor can not be final, abstract, static , synchronized.

5 Reflect

- Import java.lang.reflect.* name ka package hai.
- Yadi newInstance() method me parameter pass karna chate hai to getConstructor(int.class) method ka use karna hoga jiska return type Constructor class hai. and ye reflect package me rakhi hai.

Ex:-

```
Import java.lang.reflect.*
class A{
        public A(int x, int y){
        System.out.print("sum= "+(x+y));
        }
}
Class Demo{
 psvm(String arr[])throws Exception{
```

```
Class c=Class.forName("ClassName");
Constructor co=c.getConstructor(int.class,int.class);
Co.newInstance(100,200);
}
}
```

- Constructor public hona compulsory hai. Yadi public nhi banaya to NoSuchMethodException aa jayegi.
- Yadi humne method me parameter type difference pass kiya but constructor me different de diya to IllegalArgumentException atti hai.
- Yadi Constructor co=c.getConstructor(int.class) method ka use kiya hai to only constructor ko public hi declare ker sakte hai. Yadi **private , protected , default.** Ka use kiya to NoSuchMEthodException. ayegi . Yadi hum cahate hai ki **private , protected , default** . ko bhi access kare to getDeclaredCnstructor(int.class) ka use karna hoga.
- Hum private data ko khi bhi access ker sakte hai.Using [co.setAccessible(true);]

Q Difference between new and newInstance() method ?

Ans:-

new	newInstance()
• new is keyword.	newInstance() is method.
• Compile time pe decide karna padta hai kis class ka object create karna hai.	Runtime pe working karna ke liye newInstace() method ka use karte hai.
• new se ager hum abstract class ka object banate hai to compile time pe error aati hai.	newInstance() se ager hum abstract class ka object banate hai or runtime Exception aati hai.

Q Program to Count all method of program?

Q Program to count all variables of class?

- Yadi hum kisi bhi variable ko globely declare karte hai with final keyword to vo blank final variable ki work karta hai esliye usko initialize karna compulsory hota hai by constructor or instance block ager vo instance hai to static hai to static block se hi only initialize ker sakte hai.

Q By how many types we can create Object of class in java?

Ans:- We can create Object of class in java by 5 types.

Q Jab hum clone method se Object create karte hai to cloned Object ke corresponding Constructor call hota hai ya nhi?

Ans:- Yadi hum shallow cloning ka use ker ke karte hai to cloned Object ke corresponding Constructor call nhi hota hai. Kyo ki Constructor ka use instance variables ko initialize karne ke liye karte hai and jab first time Object create hoga to us Object ke corresponding hi instance variable initialize ho chuke hoge to cloned Object se Constructor call nhi hoga kyo ki meaning less hai. Kyo ki cloned Object to tabhi Banega jab pahle se Object created hoga. But yadi hum Deep cloning ka use ker ke karte hai to constructor cloned Object ke corresponding.

Q How we can create Object if we make the Constructor as private ?

Ans:-If we make Constructor private than we can create an Object using factory method. A factory method is method which is static and return the instance of the class.

Q Can we create without new operator in java?

Ans:-Yes , It is possible to create on Object without using new keyword in java.

- **By using clone() method.**
- **By using newInstance() method.**
- **By using serialization and deserialization.**

Q What is Pass by value and Pass by function?

Ans:-

pass by value :- A method is called pass by value if only the value part of the variable is passed as an argument to the function.

pass by address :- A method is called pass by address if the address of variable is passed as an argument.

6. Collection

Class = Green Color

Interface=Black

- **Iterable (I)**
- **Collection (I)**
  i. List (I) ii. Set (I) iii. Queue (I)
- **List (I)**
  i. ArrayList ii. LinkedList iii. Vector -> Stack
- **Set (I)**
  i.HashSet -> linkedHashSet ii.SortedSet(I) -> NevigableSet(I) ->TreeSet

- **Queue (I)**
  i. AbstractQueue -> PriorityQueue ii. BlockingQueue-> PriorityBlockingQueue ➔ LinkedBlockingQueue

Map

- **Map (I)**
  i. HashMap -> LinkedHashMap ii. SortedMap(I) -> NavigableMap(I) -> TreeMap iii. WeakHashMap iv. IdentityHashMap v. Dictionary <- HashTable -> Properties.

- Humesa Wrapper class ke case me only Corresponding Wrapper class me hi Convert hota hai. Otherwise error dega.

- Object

  i. Boolean ii. Number -> Byte -> Short -> Integer -> Long -> Float -> Double
  iii. Character iv. String

* Super class ka reference variable sub class ke object ko hold ker sakta hai.Kyo Number Super class hai. Sabhi Wapper classes ki.

* Esse ye advantage hai ki abhi har data type ke corresponding uske size ki memory hi allocate hogi like for int 4 byte for double 8 byte.

* hum Templet class and varargs ka use ek sath ker sakte hai. Like class A<T>{ A(T...x){}}

* Jab hum compile karte hai to warning aati hai But yadi hum recompile karna hoga.

Like java -xlint:unchecked Demo1.java

- But esa karne pr error aa jayegi Heap pollution Problem.
- Number of Constructor :-
  Character :- 1 Constructor
  Float :- 3 Constructor
  Others :- 2 (Byte , Short , Integer , Long ,Double , Boolean)
- "130" me nhi lika simple likha simple 130 likha to -126 aayega range me move hoga.
- Yadi Long me Integer ki range se jiyada dala or L nhi likha to Integer Number to large.(2147483648L)

Q What are the limitations of Object types Array ?

Ans:- * Object type array are of fixed size that means we can not increase or decrease the size of Object type array at runtime.

- In Object type array can be store only Homogeneous type of data.
- Insertion and deletion operation in Object type array is complex because lot of shifting is need to be done.
- Array does not contains any predefine method to search and sort.

Q What is the difference between Object type array and Collection ?

Ans:-

Array	Collection
• Size of array is fixed.	Collection is of growable type.
• According to memory point of view we doesn't prefer to use Array.	According to performance we does not prefer to use Collection over Array.
• We can search the element from array by using index.	In Collection we can not use index for search an element.

• Array can store homogeneous type of data.	Collection can store heterogenous type of data.
• There is no such method define in array for sorting and searching.	There are some methods available in Collection to sort and search the data.
• Array can store primitive datatype and Object both.	Collection can store only Object type of data.

Q What is Collection ?

Ans:- Grouping of different type of Objects into single entity is called a Collection.

(Bahot sare different type ke Object ke Collection ko single group me rakhna hi Collection kahte hai.)

Q What is Collection Framework?

Ans:- Java has provided number of classes and interface to manipulate , Store and represent the Collection. The topic of java which provide this methods classes and interfaces is called as Collection Framework.

- Kisi bhi chij means class ya interface ko parent ya root level esliye banaya jata hai taki common properties usme define ker sakte to sabhi child classes uska use ker sake.

Q What is Collection interface?

Ans:- All the common methods to insert delete and search data in Collection is defined in this interface.

- It is the root for all other interface.
- Most of the Collection framework , interface , and classes extends and imlements Collection interface.

Q What is difference between Collection and Collections ?

Ans:-

Q All the methods of Collection interface ?

Ans:- i. boolean add(Object e);

ii. boolean addAll(Collection c);
iii. void clear()
iv. boolean contains(Object o)
v. boolean containsAll(Collection c)
vi. boolean equals(Object o)
vii. int hashCode()
viii. boolean isEmpty();
ix. Iterator iterator()
x. boolean remove(java.lang.Object)
xi. boolean removeAll(Collection c)
xii. boolean retainAll(Collection c)

xiii. int size()

xiv. object[] toArray()

xv. object[] toArray(Object ar[])

Q What is List interface ?

Ans:- It is child interface of Collection interface .

- It extends Collection interface.
- It allows to store duplicate data.
- In List interface insertion order is preserved.
- In List interface we can store heterogeneous type of data.
- In List we can search data on the basis of index.

Q Methods of List interface ?

i. void add(int index , Object o)
ii. boolean addAll(int index , Collection c)
iii. Object get(int index)
iv. int indexOf(Object o)
v. int lastIndexOf(Object o)
vi. ListIterator listIterator(int index)
vii. ListIterator listIterator()
viii. Object remove(int index)
ix. Object set(int index , Object o)
x. List sublist(int start , int end)

ArrayList

- ArrayList by default 3 interface ko implements karte hai and 1 class ko extends karti hai.
- Iterator [I]
- Collection [I]
- List
  i. AbstractList <- Arraylist ->RandomAccess[I] -> Clonable[I] -> Serializable[I]

Q What is ArrayList ?

Ans:- It implements List interface and extends AbstractList class.

- ArrayList is growable in nature that's why size of ArrayList can be increase and decrease at runtime.
- It allows to store duplicate data.
- In ArrayList insertion order is preserved.
- In ArrayList insertion and deletion operation is complex.
- We prefer to use ArrayList when we need to perform search operation.
- null value can be also inserted in ArrayList.
- ArrayList can store hetrogeneouse type of Object.
- It is not synchronized.
- It is not thread safe.

- Automatically jo hum data **add(10)** method me pass karte hai vo version 1.5 ya uske bad ke version me hi work karta hai kyo ki autoboxing unboxing ka concept version 1.5 se hi aya tha. Yadi hum uske pahle vale version me chalana cahate hai to fer

add(new Integer(10)) ese likhna hoga.

- Command to run program in different version **javac -source 1.4 ProgramName.java**
- Yadi hum Wrapper class ka use karte hai and yadi direct value dalte hai to only Corresponding Wrapper class me hi dal sakte hai different me nhi dal sakte like
  Ex:- Double d=10//Error

  Double d=10.0;//Work

- But yadi hum new keyword ka use ker ke Object banate hai wrapper class ka to fer pass ker sakte hai same ya usse lowest Wrapper class ke data ko.
  Ex:- Double d=new Double(10)//Work
  Double d=new Double(10.0)//Work