1. What are static blocks and static initalizers in Java ?

Static blocks or static initializers are used to initalize static fields in java. we declare static blocks when wewant 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 thenumber 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 classand subclass then we say subclass method is overridden by superclass.

When to use overriding in javaIf 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 superclassmethod.

4. What is super keyword in java ?

Variables and methods of super class can be overridden in subclass . In case of overriding , a subclassobject call its own variables and methods. Subclass cannot access the variables and methods ofsuperclass 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 isused 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 sameclass

Method Overriding occurs betweentwo classes superclass and subclass

2) Since it involves with only one class inheritanceis not involved.

Since method overriding occurs between superclassand subclass inheritance is involved.

3) In overloading return type need not be the same

In overriding return type must be same.

4) Parameters must be different when we dooverloading

Parameters must be same.

5) Static polymorphism can be acheived usingmethod overloading

Dynamic polymorphism can be acheived usingmethod overriding.

6) In overloading one method can‟t hide theanother

In overriding subclass method hides that of thesuperclass method.

6. Difference between abstract class and interface ?

Interface :

1) Interface contains only abstract methods

2) Access Specifiers for methods in interface must be public

3) Variables defined must be public , static , final

4) Multiple Inheritance in java is implemented using interface

Abstract Class:

1) Abstract class can contain abstract methods,concrete methods or both

2) Except private we can have any accessspecifier for methods in abstract class.

3) Except private variables can have any accessspecifiers

4) We cannot achieve multiple inheritance using abstract class.

7. Why java is platform independent?

The most unique feature of java is platform independent. In any programming language soruce code iscompiled in to executable code . This cannot be run across all platforms. When javac compiles a javaprogram 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 madeavailable across all platforms by Sun Microsystems, we can execute this byte code in any platform. Bytecode generated in windows environment can also be executed in linux environment. This makes javaplatform 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 thosemethods 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 inputsor values. When an overloaded method is invoked java first checks the method name, and the number ofarguments ,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 staticbinding 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

1) Java is platform independent
2) There are no pointers in java
3) There is no operator overloading in java
4) There is garbage collection in java
5) Supports multithreading.

C++

1) C++ is platform dependent.
2) There are pointers in C++.
3) C ++ has operator overloading.
4) There is no garbage collection
5) Does‟nt support multithreading

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 ofinstructions called byte code. Byte code is a machine independent language and contains set ofinstructions 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 toaccess 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 ortemplate 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 anynumber of department objects.

All programming constructs in java reside in class. When JVM starts running it first looks for the classwhen 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 mainmethod.

14. What is an object ?

An Object is instance of class. A class defines type of object. Each object belongs to some class.Everyobject contains state and behavior. State is determined by value of attributes and behavior is calledmethod. 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 allit consists inside. He is required to know only about how to start and stop the car. So we can exposewhat 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 declaredpublic 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 thehelp of object. Then we declare that method as static. JVM calls the main() method without creatingobject by declaring keyword static.

void : void return type is used when a method does“nt return any value . main() method does“nt returnany 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 objectis created it is initialized automatically with the help of constructor in java.

We have two types of constructorsDefault ConstructorParameterized 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 instring.

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 inarray.

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. Thatis the reason, if we see C language we can write c language only in English we can‟t write in otherlanguages 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 Javasupports 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 touse 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. Characterconstants 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 createconstants 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 filledwith zeroes.

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 followjava coding standards.Classnames should start with uppercase letter. Classnames names should be nouns. If Class name is ofmultiple 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?

1) Overloading occurs when two or more methods in one class have the same method name but different parameters.

Overriding means having two methods with the same method name and parameters (i.e., method signature). One of the methods is in the parent class and the other is in the child class. Overriding allows a child class to provide a specific implementation of a method that is already provided its parent class.

2) The real object type in the run-time, not the reference variable's type, determines which overridden method is used at runtime. In contrast, reference type determines which overloaded method will be used at compile time.

3) Polymorphism applies to overriding, not to overloading.

4) Overriding is a run-time concept while overloading is a compile-time concept.

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 ofduplicating 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 EngineThe main advantage of inheritance is reusability of CodeThe main advantage of has a relationship isreusability 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 class**InstanceOfExample {**public static void**main(String[] args) {Integer a = **new**Integer(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 nota 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 canhave 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 aspublic 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.

We can‟t have more than one package statement in source file. In any java program there can be atmostonly 1 package statement. We will get compilation error if we have more than one package statement insource file.

40. What is object cloning?

The object cloning is a way to create an exact copy of an object. The clone() method of the Object class is used to clone an object. The java.lang.Cloneable interface must be implemented by the class whose object clone we want to create. If we don't implement Cloneable interface, clone() method generates CloneNotSupportedException. The clone() method is defined in the Object class. The syntax of the clone() method is as follows:

protected Object clone() throws CloneNotSupportedException

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 firststatement 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 15characters

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 ofclass, methods and members.

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

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 haveaccess modifiers and non access modifiers.

Access Modifiers : public, private, protected, defaultNon 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

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 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 accessmodifier is security no one can modify our classes, variables and methods. The main disadvantage of finalaccess 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 aclass 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 wherewe may not want to use this feature. Then we declared method as final which will print overriding. Toallow 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 valueof final variable. Any attempt to modify the final variable results in compilation error. The error is asfollows

"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 methodsin a class. We want to leave the implementation to a class that extends it. In such case we declare a classas abstract.To make a class abstract we use key word abstract. Any class that contains one or moreabstract methods is declared as abstract. If we don"t declare class as abstract which contains abstractmethods 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 twowheelers , four wheelers etc. At that moment we make vehicle class abstract. All the common features ofvehicles are declared as abstract methods in vehicle class. Any class which extends vehicle will provideits 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 superclassreferences 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 cannotinstantiate 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 withkeyword 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.

52. What is an exception in java?

In java exception is an object. Exceptions are created when an abnormal situations are arised in ourprogram. Exceptions can be created by JVM or by our application code. All Exception classes are definedin 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 anexception is raised in program it leads to termination of program when it is not handled

properly. Thesignificance of exception handling comes here in order not to terminate a program abruptly and tocontinue 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 asException, but some times exceptions are caused due to some environment issues such as running out ofmemory. In such cases we can‟t handle the exceptions. Exceptions which cannot be recovered are calledas 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 withException root class we can handle with specific exceptions. It is recommended to handle exceptions withspecific 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 thatexception is propagated or thrown to the caller of that method. This propogation continues till it finds anappropriate exception handler ,if it finds handler it would be handled otherwise program terminatesabruptly.

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 catchesthe exception thrown by preceding try block. Catch block cannot catch an exception thrown by another tryblock. If there is no exception causing code in our program or exception is not raised in our code jvmignores 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 willresult 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 morecatch clauses, each catch handling different type of exception. When an exception is thrown jvm checkseach catch statement in order and the first one which matches the type of exception is execution andremaining 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 ordershould 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 blockexecutes with no exceptions then finally is called after try block without executing catch block. If there isexception 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 tryblock.

```
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 aftercatch 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 reducesthe 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 checkedexceptions.

2) Checked exception should be thrown with keyword throws or should be provided try catch block, elsethe 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 exceptionsbecause 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 exceptionsbecause there may be many places causing exceptions.

Example :

1) Arithmetic Exception

2) ArrayIndexOutOfBoundsException

3) ClassCastException

4) IndexOutOfBoundException

5) NullPointerException

6) NumberFormatException

7 StringIndexOutOfBounds

8) UnsupportedOperationException

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

Unchecked Exception Checked Exception

1) All the subclasses of RuntimeException arecalled unchecked exception.

All subclasses of Throwable class exceptRuntimeException are called as checked exceptions

2) Unchecked exceptions need not be handled atcompile timeChecked Exceptions need to be handled at compiletime.

3) These exceptions arise mostly due to codingmistakes 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 isexception handling code then exception handles and continues the program. If there is no exceptionhandling 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 exceptionand 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 aresituations where we have to throw userdefined exceptions or runtime exceptions. In such case we usethrow 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 exceptionobject is thrown JVM checks is there any catch block to handle the exception. If not then the next catchstatement till it finds the appropriate handler. If appropriate handler is not found ,then default exceptionhandler 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 writeany 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 callermethods, 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 incompatibletypes.

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 exceptionsthrown 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 ifthere 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 statementfinally 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 handlingchecked exceptions .This raises compile time error if we try to handle checked exceptions when there isno 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 sothat 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 extendRuntime 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 exceptionfrom 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 misspelledand when we try to load the class by string name hence class cannot be found which raisesClassNotFoundException.

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. Thismight be due to misspelled classname at command line, or classpath is not specified properly , or theclass file with byte code is no longer available.

82. How to instantiate member inner class?

OuterClassName.InnerclassName inner=new OuterClassReference.new InnerClassName();

We cannot instantiate inner class without outer class reference

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 addressspace. 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 resourceswhen compared to process

4) Threads exists in process. A process have atleast one thread.

85. Difference between process and thread?

Process

1) Program in execution.

2) Processes are heavy weight

3) Processes require separate address space.

4) Interprocess communication is expensive.

5) Context switching from one process to another is costly

Thread

1) Separate path of execution in program. One ormore threads is called as process.

2) Threads are light weight.

3) Threads share same address space.

4) Interthread communication is less expensivecompared to processes.

5) 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 Usingspreadsheet 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 ofprogram. In single threaded environment each task has to be completed before proceeding to next taskmaking 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 overridingrun() we can create thread in java.

java.lang.Runnable : Runnable is an interface in java. By implementing runnable interface and overridingrun() 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 threemethods 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 thenew 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 implementingrun() 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. Inboth 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 aremultiple 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 robinfasion.

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 weneed 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 inthe 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 garbagecollected.

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 isnot 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 isstarted it will cause RunTimeException(IllegalThreadStateException). A runnable thread cannot berestarted.

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 beexecuted 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 thisstart() method.

103. 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 firstacquire a lock . If already a lock has been acquired by other it can‟t access that shared

resource. At thismoment the thread has to wait until another thread releases the lock on shared resource. To lock anobject we use synchronization in java.

A lock protects section of code allowing only one thread to execute at at a time.

104. 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.

105. What are synchronized methods ?

If we want a method of object to be accessed by single thread at a time we declare that method withsynchronized keyword.

Signature :public synchronized void methodName(){}

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

106. 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 suchscenario we can declare a method as synchronized.

107. 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.

108. 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 ofan object.

public **synchronized void** methodName()

{

}

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

109. What are synchronized blocks in java?

Synchronizing few lines of code rather than complete method with the help of synchronized keyword arecalled synchronized blocks.

Signature : Synchronized (object reference){// code}

110. 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. Themain advantage of synchronized blocks over synchronized methods is it reduces the waiting time ofthreads and improves performance of the system.

111. What is class level lock ?

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

112. 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 synchronizemethod it need to acquire first class level lock. When a thread was executing static synchronized methodno 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){}

113. Can we use synchronized block for primitives?

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

114. 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. Bydefault thread has normal priority of 5. Thread scheduler uses thread priorities to decide when eachthread is allowed to run. Thread scheduler runs higher priority threads first.

115. 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;

116. 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 thethread.

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()

{

}

117. 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 inthe pool. It depends on thread scheduler to which thread to execute. The scheduler can do any of thefollowing 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.

118. 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()

119. Explain yield() method in thread class ?

Yield() method makes the current running thread to move in to runnable state from running state givingchance to remaining threads of equal priority which are in waiting state. yield() makes current thread tosleep for a specified amount of time.There is no guarantee that moving a current running thread fromrunnable 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 loseany lock if it has acquired a lock earlier.

Signature :

```
public static native void yield()
{
}
```

120. 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 ofequal priority to execute. Thread scheduler decides whether it get chance for execution again or not. It alldepends on mercy of thread scheduler.

121. 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
{
}
```

122. Explain purpose of sleep() method in java?

sleep() method causes current running thread to sleep for specified amount of time . sleep() method isthe 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 {
}
```

123. 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 thethread sleeps for a specified amount of time.

124. Can sleep() method causes another thread to sleep?

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

125. Explain about interrupt() method of thread class ?

Thread class interrupt() method is used to interrupt current thread or another thread. It doesnot meanthe current thread to stop immediately, it is polite way of telling or requesting to continue your presentwork. 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() methodstatus would set to true. This causes the current thread to continue its work and does not have impactimmediately.

If a thread is in sleeping or waiting status (i.e thread has executed wait () or sleep() method) thread getsinterrupted it stops what it is doing and throws an interrupted exception. This is reason we need to handleinterrupted exception with throws or try/ catch block.

126. 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 theymay perform related tasks. Interthread communication in java is done with the help of following three

methods :

1) wait()

2) notify()

3) notifyAll()

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

wait() : wait() method() makes the thread current thread sleeps and releases the lock until some otherthread 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. Thehighest 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 trycatch clause.

128. 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 threadclass.

129. Explain IllegalMonitorStateException and when it will be thrown?

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

130. 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 arecalled 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.

131. 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

132. What are thread groups?

Thread Groups are group of threads and other thread groups. It is a way of grouping threads so thatactions 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 thethreads that are created belong to default thread group of the main thread. Every thread belongs to athread group. Threads that belong to a particular thread group cannot modify threads belonging toanother thread group.

133. What are thread local variables ?

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

134. What are daemon threads in java?

Daemon threads are threads which run in background. These are service threads and works for thebenefit 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 isdaemon , child thread also inherits daemon nature of thread.

135. 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 usingsetDaemon() 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 bethrown.

136. 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.

137. 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.

138. 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 nestedclasses. 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

139. 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. Thiscode is related only to that class and cannot be reused for other class . Instead of creating a new class wecan 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 variablescan 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.

140. 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 classesare 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.

141. How to instantiate static nested classes in java?

We can access static members and static methods of outer class without creating any instance of outerclass.

Syntax for instantiating Static nested class :

OuterclassName.StaticNestedClassName ref=new OuterclassName.StaticNestedClassName();

142. 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 classonly inside method where class is defined. A local inner classes exist only when method is invoked andgoes out of scope when method returns.

143. 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 localinner 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.

144. Explain about anonymous inner classes in java?

Inner class defined without any class name is called anonymous inner class. Inner class is declared andinstantiated using new keyword.The main purpose of anonymous inner classes in java are to provideinterface implementation. We use anonymous classes when we need only one instance for a class. We canuse 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

145. 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.

146. 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. Butwe are not instantiating interface we are instantiating anonymous inner class which is implementation ofRunnable interface.

147. Explain about member inner classes?

Non static class defined with in enclosing class are called member inner class. A member inner class isdefined at member level of class. A member inner class can access the members of outer class includingprivate 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.

148. How to instantiate member inner class?

OuterClassName.InnerclassName inner=new OuterClassReference.new InnerClassName();

We cannot instantiate inner class without outer class reference

149. How to do encapsulation in Java?

Make instance variables private.

Define getter and setter methods to access instance variables .

150. 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 itcan‟t point to other objects.

A reference variable can be declared either to a class type or interface type. If a reference variable isdeclared with interface type it points to the class that implements the interface.

151. Will the compiler creates a default constructor if I have a parameterized constructor inthe class?

No compiler won‟t create default constructor if there is parameterized constructor in the class.

Forexample 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 inthe Car class.

152. 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 awarning message that method name is same as class name.

153. 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 ofverriding constructors in java.

154. 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 instatic method we get compilation error. The error is as follows:

Cannot make a static reference to the non static field name

155. How do we access static members in java?

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

156. Can we override static methods in java?

Static methods can‟t be overridden. If we have a static method in superclass and subclass with samesignature then we don‟t say that as overriding. We call that as

157. 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 toaccess 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 otherreference ,passed to a method.

158. Objects or references which of them gets garbage collected?

Objects get garbage collected not its references.

159. 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 garbagecollector will call finalize() method to free the resources. Finalize() method will be called only when objectis eligible for garbage collection.

160. 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. Thelargest amount of data that can passed as parameters are long or double.

161. Explain wrapper classes in java?

Converting primitives to objects can be done with the help of wrapper classes. Prior to java 1.5 we useWrapper classes to convert primitives to objects. From java 1.5 we have a new feature autoboxing whichis used to convert automatically primitives to objects but in wrapper classes programmer has to take careof converting primitives to objects.

Wrapper classes are immutable in java. Once a value is assigned to it we cannot change the value.

162. 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

163. 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 theobject not to be saved, then we declare that variable or field as transient.

Example : public Class Car implements serializable

{

transient int carnumber;

}

164. Can we serialize static variables in java?

Static variables cannot be serialized in java.

165. 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

166. 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 longbecause long is larger datatype than int.

Widening Conversion comes under Automatic type conversion.

167. 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 donarrowing 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.

168. 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 fullyqualified class names

169. 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 departmentand project name and then the name of package.

Example : com.google.sales.employees

170. What is classpath ?

The path where our .class files are saved is referred as classpath.JVM searches for .class files by using theclass path specified. Class path is specified by using CLASSPATH environment variable. CLASSPATHenvironment variable can contain more than one value. CLASSPATH variable containing more than onevalue 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 andclasses.

171. what is jar?

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

172. What is the scope or life time of instance variables ?

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

173. 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.

174. 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 localvariables gets created in stack memory and this variable gets deleted from memory once the methodexecution is done.

175. 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 isreferred as static import. The advantage of static import is we can access static variables without class orinterface 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.*

176. Can we define static methods inside interface?

We can‟t declare static methods inside interface. Only instance methods are permitted in interfaces.onlypublic and abstract modifiers are permitted for interface methods. If we try to declare static methodsinside interface we get compilation error saying“Illegal modifier for the interface method Classname.methodName(); only public & abstract arepermitted”.

177. Define interface in java?

Interface is collection of abstract methods and constants. An interface is also defined as pure or 100percent 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. Implementskeyword is used to implement interface.

178. 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 canimplement interface. Interfaces are designed to support dynamic method resolution at run time.

179. Explain features of interfaces in java?

1) All the methods defined in interfaces are implicitly abstract even though abstract modifier is notdeclared.

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.

180. Explain enumeration in java?

Enumeration is a new feature from Java 5.0. Enumeration is set of named constants . We use enumkeyword to declare enumeration. The values defined in enumeration are enum constants.Each enumconstant 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.

181. 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 theenum constants otherwise we get compilation error.

182. Explain about field hiding in java?

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

183. Explain about Varargs in java?

Beginning with Java 5 has a new feature Varargs which allows methods to have variable number ofarguments. It simplifies creation of methods when there are more number of arguments. Earlier to java 5Varargs 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 argsis 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 arepassed.

184. Explain where variables are created in memory?

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

185. 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 usestrings in Switch statement. If we use strings in switch statement prior to Java 7 we will get compile timeerror "only int and enum constants are permitted".

186. 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. Forexample if we have a reference r1 that point to object .so when we declare r2=r1, we are assigningreference r1 to r2 so now r2 points to the same object where r1 points. Any changes done by onereference on an object will reflect to other.

187. 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 orfunctions are determined next they concentrate on storing data.

Features :

1) In this top down approach is followed. First procedures were determined and then concentrate onminute details.

2) Concentrate more on functions and procedure rather than data.

3) In traditional programming language procedures manipulate global data without knowing to otherprocedures.

4) Very little concentration on minute detailsThe main drawback of traditional programming languages works well only for small problems. But notsuitable for larger problems.

Ex : C language, Pascal

188. 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 thenconcentrates 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.

189. List out benefits of object oriented programming language?

1) Easy maintenance

2) Code reusability

3) Code extendability

4) Reliable

190. Differences between traditional programming language and object oriented programminglanguage?

Traditional Programming language Object Oriented Programming LanguageA 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 noencapsulation all procedures access data.

In oops encapsulation is done by tightly couplingdata and behaviour together in class.

Suitable for small programs or problems Suitable for large programs and complex problems.

191. Explain oops concepts in detail?

Object oriented programming should support these three features :

1) Inheritance

2) Encapsulation

3) Polymorphism

192. Explain what is encapsulation?

Encapsulation is the process of wrapping of code and behaviour in a single unit called class and preventingfrom misuse is called encapsulation. Encapsulation exposes only part of object which are safe to exposedand 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 whichit is made up of.

193. What is inheritance ?

Inheritance is one of the important feature of object oriented language. Inheriting is the process ofacquiring 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 iscalled 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 sinceobject class is the super class for all classes in java.

When we create a new class by using inheritance „is-a‟ relationship is formed.

194. Explain importance of inheritance in java?

Reusability :The major advantage of inheritance is code reuse. We can avoid duplicating code by usinginheritance. We can place all common state and behaviour in that class , by extending that class we canExtendability : 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.

195. What is polymorphism in java?

Polymorphism is combination of two greek words which mean many forms. In polymorphism actual typeof object involved in method call determines which method to call rather type of reference variable.

196. What is covariant return ?

In java 1.4 and earlier one method can override super class method if both methods have same signatureand return types.

From Java 1.5 , a method can override other method if argument types match exactly though returntypes 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 andearlier.

197. What is collections framework ?

A framework is set of classes and interfaces to build a functionality. Java collections framework providesset of interfaces and classes for storing and manipulating collections. Collection framework containsclasses 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 ofuseful data structures and algorithms.

198. 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.

199. 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.

200. 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(Objecto);

Removes an object from collection if that object is present in collection. Returntrue if matching object is removed from collection.

booleanaddAll(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(Objecto);

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 onlyelements specified in collection removing other elements.

Object[] toArray(); Returns an array of elements in collection.

201. List the interfaces which extends collection interface ?

1) List

2) Set

3) Queue

4) Deque (From Java 6)

202. 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.

203. 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 specifiedposition in the list.

E set(int index, E element); This method replaces the element at specifiedposition in the list with the specified element.

void add(int index, E element); This method inserts the specified element with theindex specified.

E remove(int index); This method removes the element at specifiedindex and returns the element removed.

int indexOf(Object o); indexOf() method returns the index of lastoccurrence of specified element. If there is noelement 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 betweenindexes specified.

204. List implementations of List Interface ?

1) ArrayList

2) Vector

3) LinkedList

205. 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.

206. 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

1) While creating array we have to know the size.

2) Arrays are static

3) We can store objects and primitives

4) We have to manually write logic for inserting and removing elements.

5) Arrays are faster

ARRAY LIST

1) But it is not required to know size whilecreating ArrayList, because arraylist growsdynamically.

2) ArrayList is dynamic

3) We can store only primitives prior to 1.5 . From1.5 we can store even objects also.

4) Just a method call would add or removeelements from list.

5) Arraylist is slower.

207. 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

208. 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 issynchronized which reduces performance if only one thread accesses it.

209. 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 elementsin 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 tomove each element to next position and insert the element. Similarly with remove if we want to removean 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 theneighbouring node reference.

Linked list class is in java.util package.

Linked List class extends class extends AbstractSequentialList and I mplements List, Deque, Cloneableand 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.

210. 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 toaccess elements one by one in collection. Iterator is an object associated with collection used to loopthrough 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 callingnext() first we should call hasNext() whether it has elements or not. If there is next element we can callnext() so that we can avoid exception.

211. 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, andfor 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 someadditional methods .

List Iterator doesn‟t have current element .Position of List Iterator lies between two elements i.e previouselement 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 ifthere are no elements.

E next(); Returns the next element . NoSuchElementException is thrown if there is no nextelement.

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.

intpreviousIndex();

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 oflist.

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 whichdoesn‟t allow duplicates.Adding an duplicate element to a set would be ignored .Set interface isimplemented in java.util.set package.Set interface does not have any additional methods . It has onlycollection 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 comingto 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 treefor 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.

218. 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.

219. 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"tmaintain order. Maintaining order means elements will be retrieved in order which they are inserted.

220. 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.

221. 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. Linkedhashmap is preferred only for faster iteration of elements.

public class LinkedHashMap<K,V>

extends HashMap<K,V>

implements Map<K,V>

{

}

222. 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 usingcomparator.

public interface SortedMap<K,V> extends Map<K,V> {

}

223. 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**.

224. Difference between HashMap and Hashtable?

Difference HashMap Hashtable

1) Synronization HashMap is **not synchronized**. Hashtable is **synchronized**.

2) Nulls HashMap allows atmost **one null keyand any number of null values**.

Hashtable **does not allow null values**.

3) Performance Since HashMap is not synchronized itsperformance is **faster than** Hashtable.

Performance is **slower** when compared toHashMap.

4) Introduction HashMap introduced starting from Hashtable is even before collectioncollection framework.

225. Difference between arraylist and linkedlist?

Difference Arraylist& LinkedList

Access Implements RandomAccess interface wecan **search randomly** all the elements in the list.

It extends Abstract sequential Listinterface which provides**sequential access** to elements.Searching and retrieval ofelements**Searching and retrieval** of elements is**fast** since arraylist provides randomaccess.

Searching and retrieval ofelements is **slow** because ofsequential access to elements.

Addition and removal ofelements**Adding and removal of elements inrandom positions is slow**.

For example: if we want to add element to middle ofthe list we have to move the elements inthe list and then we need to insert theelement. Similarly for removing theelement we need to follow the samething.

Adding and removal ofelements in random positionsis fast because there is no need ofresizing the array just by updatingthe node structures with newaddresses.

226. Difference between Comparator and Comparable in java?

Comparator

1. Comparator Defined in java.util package.
2. Comparator interface is used when we want to compare two different instances.
3. Comparator is used when we want **custom sorting**.
4. Should override int compare(T o1, To2) method which takes two instances.
5. For sorting objects we use collections.sort(list,new Comparator);

Comparable

1. Defined in **java.lang package.**
2. Comparable is used **to** compare itself with otherinstance.
3. Comparable is used for natural sortingof objects.
4. Should override public into compareTo(T o) methodwhich takes one instance.
5. For sorting objects we use collections.sort(list);

227. 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.

228. Difference between Concurrent HashMap and Hashtable andcollections.synchronizedHashMap?

Locking Mechansim :ConcurrentHashMap uses completely different hashing mechanism called lockstriping which offers better concurrency and scalability.The main advantage of this mechanism is better concurrency instead of synchronizing every method byusing common lock which allows only one thread to access at a time, it allows better concurrency byallowing multiple threads to access.

ConcurrentModificationException :ConcurrentHashMap provides iterators which doesnot throw concurrentmodification exception which allows only one thread to access iterator, while synchronized map maythrow concurrent modification exception.

229. Explain copyOnWriteArrayList and when do we use copyOnWriteArrayList?

copyOnWriteArrayList is used in multithreaded environment. If we want to iterate over arraylist ,but thearraylist 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) anditerate over the list in synchronized block to avoid concurrent modification exception.

2) The second one is to use copyOnWriteArrayList which takes care of concurrency.44The advantage of using copyOnWriteArrayList is no need to synchronize list explicitly. So when we usecopyOnWriteArrayList when a thread modifies the list while the other thread was iterating it does notmodify original list but creates a copy of list with modified contents so that the iterator won‟t know themodifications made to original list.

230. 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 suchmodification happens a concurrent modification exception will be thrown.Such kind of iterators are fail fastiterators.

Ex : ArrayList,HashSet,HashMap. Almost all the iterators implemented in collections framework are failfast.

231. Explain about fail safe iterators in java?

Fail safe iterators are iterators which does not throw concurrent modification exception, when one threadmodifies 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 notmodify original list but creates a copy of list with modified contents so that the iterator won‟t know themodifications made to original list.

Ex : copyOnWriteArrayList

232. What is serialization in java?

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

233. 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.

234. 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 andtransfer data and then unmarshall and convert to object.

JSON based transfer : We can use json to transfer data.

235. Explain about serializable interface in java?

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

Syntax:

```
public interface Serializable {
}
```

236. How to make object serializable in java?

1) Our class must implement serializable interface.If our object contains other objects those class mustalso 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

237. 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 classname,super interfaces and member. Suid is a unique id no two classes will have same suid. Whenever anobject 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 notmatch with suid read from object stream, readObject throws an InvalidClassException.

238. 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 suidrather than JVM creating because at run time JVM has to compute the hashcode of all the properties ofclass. This process makes serialization low. We can‟t serialize static fields one exception to this is suidwhere suid gets serialized along with the object.

Ex :private static final long serialVersionUID = -588556809444428487 5L;

239. Can we serialize static variables in java?

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

240. 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 referenceobjects also get serialized. If we don‟t make reference objects serializable then we get NotSerializableException.

241. 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 forobject references.

242. What is Java?

Java is the high-level, object-oriented, robust, secure programming language, platform-independent, high performance, Multithreaded, and portable programming language. It was developed by **James Gosling** in June 1991. It can also be known as the platform as it provides its own JRE and API.

243. List the features of Java Programming language.

There are the following features in Java Programming Language.

- Simple
- Object-Oriented:
- Portable

- Platform Independent
- Secured
- Robust
- Architecture Neutral
- Interpreted
- High Performance
- Multithreaded
- Distributed
- Dynamic

244. What do you understand by Java virtual machine?

Java Virtual Machine is a virtual machine that enables the computer to run the Java program. JVM acts like a run-time engine which calls the main method present in the Java code. JVM is the specification which must be implemented in the computer system. The Java code is compiled by JVM to be a Bytecode which is machine independent and close to the native code.

245. What is the difference between JDK, JRE, and JVM?

JVM

JVM is an acronym for Java Virtual Machine; it is an abstract machine which provides the runtime environment in which Java bytecode can be executed. It is a specification which specifies the working of Java Virtual Machine. Its implementation has been provided by Oracle and other companies. Its implementation is known as JRE.

JVMs are available for many hardware and software platforms (so JVM is platform dependent). It is a runtime instance which is created when we run the Java class. There are three notions of the JVM: specification, implementation, and instance.

JRE

JRE stands for Java Runtime Environment. It is the implementation of JVM. The Java Runtime Environment is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It physically exists. It contains a set of libraries + other files that JVM uses at runtime.

JDK

JDK is an acronym for Java Development Kit. It is a software development environment which is used to develop Java applications and applets. It physically exists. It contains JRE +

development tools. JDK is an implementation of any one of the below given Java Platforms released by Oracle Corporation:

- Standard Edition Java Platform
- Enterprise Edition Java Platform
- Micro Edition Java Platform

246. How many types of memory areas are allocated by JVM?

Many types:

- **Class (Method) Area:** Class Area stores per-class structures such as the runtime constant pool, field, method data, and the code for methods.
- **Heap:** It is the runtime data area in which the memory is allocated to the objects
- **Stack:** Java Stack stores frames. It holds local variables and partial results, and plays a part in method invocation and return. Each thread has a private JVM stack, created at the same time as the thread. A new frame is created each time a method is invoked. A frame is destroyed when its method invocation completes.
- **Program Counter Register:** PC (program counter) register contains the address of the Java virtual machine instruction currently being executed.
- **Native Method Stack:** It contains all the native methods used in the application.

247. What is JIT compiler?

Just-In-Time (JIT) compiler: It is used to improve the performance. JIT compiles parts of the bytecode that have similar functionality at the same time, and hence reduces the amount of time needed for compilation. Here the term "compiler" refers to a translator from the instruction set of a Java virtual machine (JVM) to the instruction set of a specific CPU.

248. What is the platform?

A platform is the hardware or software environment in which a piece of software is executed. There are two types of platforms, software-based and hardware-based. Java provides the software-based platform.

249. What are the main differences between the Java platform and other platforms?

There are the following differences between the Java platform and other platforms.

Java is the software-based platform whereas other platforms may be the hardware platforms or software-based platforms.

Java is executed on the top of other hardware platforms whereas other platforms can only have the hardware components.

250. What gives Java its 'write once and run anywhere' nature?

The bytecode. Java compiler converts the Java programs into the class file (Byte Code) which is the intermediate language between source code and machine code. This bytecode is not platform specific and can be executed on any computer.

251. What is classloader?

Classloader is a subsystem of JVM which is used to load class files. Whenever we run the java program, it is loaded first by the classloader. There are three built-in classloaders in Java.

Bootstrap ClassLoader: This is the first classloader which is the superclass of Extension classloader. It loads the rt.jar file which contains all class files of Java Standard Edition like java.lang package classes, java.net package classes, java.util package classes, java.io package classes, java.sql package classes, etc.

Extension ClassLoader: This is the child classloader of Bootstrap and parent classloader of System classloader. It loads the jar files located inside $JAVA_HOME/jre/lib/ext directory.

System/Application ClassLoader: This is the child classloader of Extension classloader. It loads the class files from the classpath. By default, the classpath is set to the current directory. You can change the classpath using "-cp" or "-classpath" switch. It is also known as Application classloader.

252. Is delete, next, main, exit or null keyword in java?

No.

253. If I don't provide any arguments on the command line, then what will the value stored in the String array passed into the main() method, empty or NULL?

It is empty, but not null.

254. What if I write static public void instead of public static void?

The program compiles and runs correctly because the order of specifiers doesn't matter in Java.

255. What is the default value of the local variables?

The local variables are not initialized to any default value, neither primitives nor object references.

256. What are the various access specifiers in Java?

In Java, access specifiers are the keywords which are used to define the access scope of the method, class, or a variable. In Java, there are four access specifiers given below.

Public: The classes, methods, or variables which are defined as public, can be accessed by any class or method.

Protected :Protected can be accessed by the class of the same package, or by the sub-class of this class, or within the same class.

Default :Default are accessible within the package only. By default, all the classes, methods, and variables are of default scope.

Private: The private class, methods, or variables defined as private can be accessed within the class only.

257. What is the purpose of static methods and variables?

The methods or variables defined as static are shared among all the objects of the class. The static is the part of the class and not of the object. The static variables are stored in the class area, and we do not need to create the object to access such variables. Therefore, static is used in the case, where we need to define variables or methods which are common to all the objects of the class.

258. What are the advantages of Packages in Java?

- There are various advantages of defining packages in Java.
- Packages avoid the name clashes.
- The Package provides easier access control.
- We can also have the hidden classes that are not visible outside and used by the package.
- It is easier to locate the related classes.

259. What is an object?

The Object is the real-time entity having some state and behavior. In Java, Object is an instance of the class having the instance variables as the state of the object and the methods as the behavior of the object. The object of a class can be created by using the new keyword.

260. What is the difference between an object-oriented programming language and object-based programming language?

There are the following basic differences between the object-oriented language and object-based language.

Object-oriented languages follow all the concepts of OOPs whereas, the object-based language doesn't follow all the concepts of OOPs like inheritance and polymorphism.

Object-oriented languages do not have the inbuilt objects whereas Object-based languages have the inbuilt objects, for example, JavaScript has window object.

Examples of object-oriented programming are Java, C#, Smalltalk, etc. whereas the examples of object-based languages are JavaScript, VBScript, etc.

261. What will be the initial value of an object reference which is defined as an instance variable?

All object references are initialized to null in Java.

262. What is the constructor?

The constructor can be defined as the special type of method that is used to initialize the state of an object. It is invoked when the class is instantiated, and the memory is allocated for the object. Every time, an object is created using the new keyword, the default constructor of the class is called. The name of the constructor must be similar to the class name. The constructor must not have an explicit return type.

263. How many types of constructors are used in Java?

Based on the parameters passed in the constructors, there are two types of constructors in Java.

Default Constructor: default constructor is the one which does not accept any value. The default constructor is mainly used to initialize the instance variable with the default values. It can also be used for performing some useful task on object creation. A default constructor is invoked implicitly by the compiler if there is no constructor defined in the class.

Parameterized Constructor: The parameterized constructor is the one which can initialize the instance variables with the given values. In other words, we can say that the constructors which can accept the arguments are called parameterized constructors.

264. Does constructor return any value?

yes, The constructor implicitly returns the current instance of the class (You can't use an explicit return type with the constructor).

265. Is constructor inherited?

No, The constructor is not inherited.

266. Can you make a constructor final?

No, the constructor can't be final.

267. What do you understand by copy constructor in Java?

There is no copy constructor in java. However, we can copy the values from one object to another like copy constructor in C++.

There are many ways to copy the values of one object into another in java. They are:

By constructor

By assigning the values of one object into another

By clone() method of Object class.

268. What are the differences between the constructors and methods?

There are many differences between constructors and methods. They are given below.

Java Constructor

- A constructor is used to initialize the state of an object.
- A constructor must not have a return type.
- The constructor is invoked implicitly.
- The Java compiler provides a default constructor if you don't have any constructor in a class.
- The constructor name must be same as the class name.

Java Method

- A method is used to expose the behavior of an object.
- A method must have a return type.
- The method is invoked explicitly.
- The method is not provided by the compiler in any case.
- The method name may or may not be same as class name.

269. What is the static method?

A static method belongs to the class rather than the object.

There is no need to create the object to call the static methods.

A static method can access and change the value of the static variable.

270. What are the restrictions that are applied to the Java static methods?

Two main restrictions are applied to the static methods.

The static method can not use non-static data member or call the non-static method directly.

this and super cannot be used in static context as they are non-static.

271. Why is the main method static?

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

272. Can we override the static methods?

No, we can't override static methods.

273. What is the static block?

Static block is used to initialize the static data member. It is executed before the main method, at the time of class loading.

274. Can we execute a program without main() method?

Yes, one of the ways to execute the program without the main method is using static block.

275. **What if the static modifier is removed from the signature of the main method?**

Program compiles. However, at runtime, It throws an error "NoSuchMethodError."

276. What is the difference between static (class) method and instance method?

class method

- A method that is declared as static is known as the static method.
- We don't need to create the objects to call the static methods.
- Non-static (instance) members cannot be accessed in the static context (static method, static block, and static nested class) directly.
- For example: public static int cube(int n){ return n*n*n;}

instance method

- A method that is not declared as static is known as the instance method.
- The object is required to call the instance methods.
- Static and non-static variables both can be accessed in instance methods.
- For example: public void msg(){...}.

277. Can we make constructors static?

As we know that the static context (method, block, or variable) belongs to the class, not the object. Since Constructors are invoked only when the object is created, there is no sense to make the constructors static. However, if you try to do so, the compiler will show the compiler error.

278. Can we make the abstract methods static in Java?

In Java, if we make the abstract methods static, It will become the part of the class, and we can directly call it which is unnecessary. Calling an undefined method is completely useless therefore it is not allowed.

279. Can we declare the static variables and methods in an abstract class?

Yes, we can declare static variables and methods in an abstract method. As we know that there is no requirement to make the object to access the static context, therefore, we can access the static context declared inside the abstract class by using the name of the abstract class.

280. What is this keyword in java?

The this keyword is a reference variable that refers to the current object. There are the various uses of this keyword in Java. It can be used to refer to current class properties such as instance methods, variable, constructors, etc. It can also be passed as an argument into the methods or constructors. It can also be returned from the method as the current class instance.

281. What are the main uses of this keyword?

There are the following uses of this keyword.

- this can be used to refer to the current class instance variable.
- this can be used to invoke current class method (implicitly)
- this() can be used to invoke the current class constructor.
- this can be passed as an argument in the method call.
- this can be passed as an argument in the constructor call.
- this can be used to return the current class instance from the method.

282. Can we assign the reference to this variable?

No, this cannot be assigned to any value because it always points to the current class object and this is the final reference in Java. However, if we try to do so, the compiler error will be shown.

283. Can this keyword be used to refer static members?

Yes, It is possible to use this keyword to refer static members because this is just a reference variable which refers to the current class object. However, as we know that, it is unnecessary to access static variables through objects, therefore, it is not the best practice to use this to refer static members.

284. How can constructor chaining be done using this keyword?

Constructor chaining enables us to call one constructor from another constructor of the class with respect to the current class object. We can use this keyword to perform constructor chaining within the same class.

285. What are the advantages of passing this into a method instead of the current class object itself?

As we know, that this refers to the current class object, therefore, it must be similar to the current class object. However, there can be two main advantages of passing this into a method instead of the current class object.

- this is a final variable. Therefore, this cannot be assigned to any new value whereas the current class object might not be final and can be changed.
- this can be used in the synchronized block.

286. What is the Inheritance?

Inheritance is a mechanism by which one object acquires all the properties and behavior of another object of another class. It is used for Code Reusability and Method Overriding. The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also. Inheritance represents the IS-A relationship which is also known as a parent-child relationship.

There are five types of inheritance in Java.

- Single-level inheritance
- Multi-level inheritance
- Multiple Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

Multiple inheritance is not supported in Java through class.

287. Why is Inheritance used in Java?

There are various advantages of using inheritance in Java that is given below.

- Inheritance provides code reusability. The derived class does not need to redefine the method of base class unless it needs to provide the specific implementation of the method.
- Runtime polymorphism cannot be achieved without using inheritance.

- We can simulate the inheritance of classes with the real-time objects which makes OOPs more realistic.
- Inheritance provides data hiding. The base class can hide some data from the derived class by making it private.
- Method overriding cannot be achieved without inheritance. By method overriding, we can give a specific implementation of some basic method contained by the base class.

288. Which class is the superclass for all the classes?

The **object class** is the superclass of all other classes in Java.

289. Why is multiple inheritance not supported in java?

To reduce the complexity and simplify the language, multiple inheritance is not supported in java.

290. What is aggregation?

Aggregation can be defined as the relationship between two classes where the aggregate class contains a reference to the class it owns. Aggregation is best described as a has-a relationship.

291. What is composition?

Holding the reference of a class within some other class is known as composition. When an object contains the other object, if the contained object cannot exist without the existence of container object, then it is called composition. In other words, we can say that composition is the particular case of aggregation which represents a stronger relationship between two objects.

292. What is the difference between aggregation and composition?

Aggregation represents the weak relationship whereas composition represents the strong relationship. For example, the bike has an indicator (aggregation), but the bike has an engine (composition).

293. Why does Java not support pointers?

The pointer is a variable that refers to the memory address. They are not used in Java because they are unsafe(unsecured) and complex to understand.

294. What is super in java?

The super keyword in Java is a reference variable that is used to refer to the immediate parent class object. Whenever you create the instance of the subclass, an instance of the parent class is created implicitly which is referred by super reference variable. The super() is called in the class constructor implicitly by the compiler if there is no super or this.

295. What are the main uses of the super keyword?

There are the following uses of super keyword.

- super can be used to refer to the immediate parent class instance variable.
- super can be used to invoke the immediate parent class method.
- super() can be used to invoke immediate parent class constructor.

296. What are the differences between this and super keyword?

There are the following differences between this and super keyword.

- The super keyword always points to the parent class contexts whereas this keyword always points to the current class context.
- The super keyword is primarily used for initializing the base class variables within the derived class constructor whereas this keyword primarily used to differentiate between local and instance variables when passed in the class constructor.
- The super and this must be the first statement inside constructor otherwise the compiler will throw an error.

297. Can you use this() and super() both in a constructor?

No, because this() and super() must be the first statement in the class constructor.

298. What is object cloning?

The object cloning is used to create the exact copy of an object. The clone() method of the Object class is used to clone an object. The java.lang.Cloneable interface must be implemented by the class whose object clone we want to create. If we don't implement Cloneable interface, clone() method generates CloneNotSupportedException.

299. What is method overloading?

Method overloading is the polymorphism technique which allows us to create multiple methods with the same name but different signature. We can achieve method overloading in two ways.

- Changing the number of arguments
- Changing the return type

Method overloading increases the readability of the program. Method overloading is performed to figure out the program quickly.

300. Why is method overloading not possible by changing the return type in java?

In Java, method overloading is not possible by changing the return type of the program due to avoid the ambiguity.

301. Can we overload the methods by making them static?

No, We cannot overload the methods by just applying the static keyword to them(number of parameters and types are the same).

302. Can we overload the main() method?

Yes, we can have any number of main methods in a Java program by using method overloading.

303. What is method overriding:

If a subclass provides a specific implementation of a method that is already provided by its parent class, it is known as Method Overriding. It is used for runtime polymorphism and to implement the interface methods.

Rules for Method overriding

- The method must have the same name as in the parent class.
- The method must have the same signature as in the parent class.
- Two classes must have an IS-A relationship between them.

304. Can we override the static method?

No, you can't override the static method because they are the part of the class, not the object.

305. Why can we not override static method?

It is because the static method is the part of the class, and it is bound with class whereas instance method is bound with the object, and static gets memory in class area, and instance gets memory in a heap.

306. Can we override the overloaded method?

Yes.

307. Difference between method Overloading and Overriding.

Method Overloading

- Method overloading increases the readability of the program.
- Method overloading occurs within the class.
- In this case, the parameters must be different.

Method Overriding

- Method overriding provides the specific implementation of the method that is already provided by its superclass.
- Method overriding occurs in two classes that have IS-A relationship between them.
- In this case, the parameters must be the same.

308. Can we override the private methods?

No, we cannot override the private methods because the scope of private methods is limited to the class and we cannot access them outside of the class.

309. Can we change the scope of the overridden method in the subclass?

Yes, we can change the scope of the overridden method in the subclass. However, we must notice that we cannot decrease the accessibility of the method. The following point must be taken care of while changing the accessibility of the method.

- The private can be changed to protected, public, or default.
- The protected can be changed to public or default.
- The default can be changed to public.
- The public will always remain public.

310. Can we modify the throws clause of the superclass method while overriding it in the subclass?

Yes, we can modify the throws clause of the superclass method while overriding it in the subclass. However, there are some rules which are to be followed while overriding in case of exception handling.

- If the superclass method does not declare an exception, subclass overridden method cannot declare the checked exception, but it can declare the unchecked exception.
- If the superclass method declares an exception, subclass overridden method can declare same, subclass exception or no exception but cannot declare parent exception.

311. Can you have virtual functions in Java?

Yes, all functions in Java are virtual by default.

312. What is the final variable?

In Java, the final variable is used to restrict the user from updating it. If we initialize the final variable, we can't change its value.

313. What is the final method?

If we change any method to a final method, we can't override it. More Details.

```
class Bike{
  final void run(){System.out.println("running");}
}
class Honda extends Bike{
```

```
  void run(){System.out.println("running safely with 100kmph");}

  public static void main(String args[]){
  Honda honda= new Honda();
  honda.run();
  }
}
```

Output:Compile Time Error

314. What is the final class?

If we make any class final, we can't inherit it into any of the subclasses.

```
final class Bike{}
class Honda1 extends Bike{
 void run(){System.out.println("running safely with 100kmph");}
 public static void main(String args[]){
 Honda1 honda= new Honda1();
 honda.run();
 }
}
```

Output:Compile Time Error

315. What is the final blank variable?

A final variable, not initialized at the time of declaration, is known as the final blank variable. We can't initialize the final blank variable directly. Instead, we have to initialize it by using the class constructor. It is useful in the case when the user has some data which must not be changed by others.

316. Can we initialize the final blank variable?

Yes, if it is not static, we can initialize it in the constructor. If it is static blank final variable, it can be initialized only in the static block.

317. Can you declare the main method as final?

Yes, We can declare the main method as public static final void main(String[] args){}.

318. Can we declare a constructor as final?

The constructor can never be declared as final because it is never inherited. Constructors are not ordinary methods; therefore, there is no sense to declare constructors as final. However, if you try to do so, The compiler will throw an error.

319. Can we declare an interface as final?

No, we cannot declare an interface as final because the interface must be implemented by some class to provide its definition. Therefore, there is no sense to make an interface final. However, if you try to do so, the compiler will show an error.

320. What is the difference between the final method and abstract method?

The main difference between the final method and abstract method is that the abstract method cannot be final as we need to override them in the subclass to give its definition.

321. What is the difference between compile-time polymorphism and runtime polymorphism?

There are the following differences between compile-time polymorphism and runtime polymorphism.

Compile-time polymorphism

- In compile-time polymorphism, call to a method is resolved at compile-time.
- It is also known as static binding, early binding, or overloading.
- Overloading is a way to achieve compile-time polymorphism in which, we can define multiple methods or constructors with different signatures.
- It provides fast execution because the type of an object is determined at compile-time.
- Compile-time polymorphism provides less flexibility because all the things are resolved at compile-time.

Runtime polymorphism

- In runtime polymorphism, call to an overridden method is resolved at runtime.
- It is also known as dynamic binding, late binding, overriding, or dynamic method dispatch.
- Overriding is a way to achieve runtime polymorphism in which, we can redefine some particular method or variable in the derived class. By using overriding, we can give some specific implementation to the base class properties in the derived class.
- It provides slower execution as compare to compile-time because the type of an object is determined at run-time.

- Run-time polymorphism provides more flexibility because all the things are resolved at runtime.

322. Can you achieve Runtime Polymorphism by data members?

No, because method overriding is used to achieve runtime polymorphism and data members cannot be overridden. We can override the member functions but not the data members.

323. What is the difference between static binding and dynamic binding?

In case of the static binding, the type of the object is determined at compile-time whereas, in the dynamic binding, the type of the object is determined at runtime.

324. What is Java instanceOf operator?

The instanceof in Java is also known as type comparison operator because it compares the instance with type. It returns either true or false. If we apply the instanceof operator with any variable that has a null value, it returns false.

325. What is the abstraction?

Abstraction is a process of hiding the implementation details and showing only functionality to the user. It displays just the essential things to the user and hides the internal information, for example, sending SMS where you type the text and send the message. You don't know the internal processing about the message delivery. Abstraction enables you to focus on what the object does instead of how it does it. Abstraction lets you focus on what the object does instead of how it does it.

In Java, there are two ways to achieve the abstraction.

- Abstract Class
- Interface

326. What is the difference between abstraction and encapsulation?

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

327. What is the abstract class?

A class that is declared as abstract is known as an abstract class. It needs to be extended and its method implemented. It cannot be instantiated. It can have abstract methods, non-abstract methods, constructors, and static methods. It can also have the final methods which will force the subclass not to change the body of the method.

328. Can there be an abstract method without an abstract class?

No, if there is an abstract method in a class, that class must be abstract.

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

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

330. Is it possible to instantiate the abstract class?

No, the abstract class can never be instantiated even if it contains a constructor and all of its methods are implemented.

331. What is the interface?

The interface is a blueprint for a class that has static constants and abstract methods. It can be used to achieve full abstraction and multiple inheritance. It is a mechanism to achieve abstraction. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in Java. In other words, you can say that interfaces can have abstract methods and variables. Java Interface also represents the IS-A relationship. It cannot be instantiated just like the abstract class. However, we need to implement it to define its methods. Since Java 8, we can have the default, static, and private methods in an interface.

332. Can you declare an interface method static?

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

333. Can the Interface be final?

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

334. What is a marker interface?

A Marker interface can be defined as the interface which has no data member and member functions. For example, Serializable, Cloneable are marker interfaces. The marker interface can be declared as follows.

```
public interface Serializable{
}
```

335. Can we define private and protected modifiers for the members in interfaces?

No, they are implicitly public.

336. When can an object reference be cast to an interface reference?

An object reference can be cast to an interface reference when the object implements the referenced interface.

337. How to make a read-only class in Java?

A class can be made read-only by making all of the fields private. The read-only class will have only getter methods which return the private property of the class to the main method. We cannot modify this property because there is no setter method available in the class.

338. How to make a write-only class in Java?

A class can be made write-only by making all of the fields private. The write-only class will have only setter methods which set the value passed from the main method to the private fields. We cannot read the properties of the class because there is no getter method in this class.

339. What are the advantages of Encapsulation in Java?

There are the following advantages of Encapsulation in Java?

- By providing only the setter or getter method, you can make the class read-only or write-only. In other words, you can skip the getter or setter methods.
- It provides you the control over the data. Suppose you want to set the value of id which should be greater than 100 only, you can write the logic inside the setter method. You can write the logic not to store the negative numbers in the setter methods.
- It is a way to achieve data hiding in Java because other class will not be able to access the data through the private data members.
- The encapsulate class is easy to test. So, it is better for unit testing.
- The standard IDE's are providing the facility to generate the getters and setters. So, it is easy and fast to create an encapsulated class in Java.

340. What is the package?

A package is a group of similar type of classes, interfaces, and sub-packages. It provides access protection and removes naming collision. The packages in Java can be categorized into two forms, inbuilt package, and user-defined package. There are many built-in packages such as Java, lang, awt, javax, swing, net, io, util, sql, etc.

341. What are the advantages of defining packages in Java?

By defining packages, we can avoid the name conflicts between the same class names defined in different packages. Packages also enable the developer to organize the similar classes more

effectively. For example, one can clearly understand that the classes present in java.io package are used to perform io related operations.

342. How to create packages in Java?

If you are using the programming IDEs like Eclipse, NetBeans, MyEclipse, etc. click on file->new->project and eclipse will ask you to enter the name of the package. It will create the project package containing various directories such as src, etc. If you are using an editor like notepad for java programming, use the following steps to create the package.

- Define a package package_name. Create the class with the name class_name and save this file with your_class_name.java.
- Now compile the file by running the following command on the terminal.

javac -d . your_class_name.java

The above command creates the package with the name package_name in the present working directory.

- Now, run the class file by using the absolute class file name, like following.

java package_name.class_name

343. How can we access some class in another class in Java?

There are two ways to access a class in another class.

- By using the fully qualified name: To access a class in a different package, either we must use the fully qualified name of that class, or we must import the package containing that class.
- By using the relative path, We can use the path of the class that is related to the package that contains our class. It can be the same or subpackage.

344. Do I need to import java.lang package any time? Why?

No. It is by default loaded internally by the JVM.

345. Can I import same package/class twice? Will the JVM load the package twice at runtime?

One can import the same package or the same class multiple times. Neither compiler nor JVM complains about it. However, the JVM will internally load the class only once no matter how many times you import the same class.

346. What is the static import?

By static import, we can access the static members of a class directly, and there is no to qualify it with the class name.

347. How many types of exception can occur in a Java program?

There are mainly two types of exceptions: checked and unchecked. Here, an error is considered as the unchecked exception. According to Oracle, there are three types of exceptions:

- Checked Exception: Checked exceptions are the one which are checked at compile-time. For example, SQLException, ClassNotFoundException, etc.
- Unchecked Exception: Unchecked exceptions are the one which are handled at runtime because they can not be checked at compile-time. For example, ArithmaticException,NullPointerException, ArrayIndexOutOfBoundsException, etc.
- Error: Error cause the program to exit since they are not recoverable. For Example, OutOfMemoryError, AssertionError, etc.

348. What is Exception Handling?

Exception Handling is a mechanism that is used to handle runtime errors. It is used primarily to handle checked exceptions. Exception handling maintains the normal flow of the program. There are mainly two types of exceptions: checked and unchecked. Here, the error is considered as the unchecked exception.

349. What is the difference between Checked Exception and Unchecked Exception?

1) Checked Exception

The classes that extend Throwable class except RuntimeException and Error are known as checked exceptions, e.g., IOException, SQLException, etc. Checked exceptions are checked at compile-time.

2) Unchecked Exception

The classes that extend RuntimeException are known as unchecked exceptions, e.g., ArithmeticException, NullPointerException, etc. Unchecked exceptions are not checked at compile-time.

350. What is the base class for Error and Exception?

The Throwable class is the base class for Error and Exception.

351. Is it necessary that each try block must be followed by a catch block?

It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block OR a finally block. So whatever exceptions are likely to be thrown should be declared in the throws clause of the method.

352. What is finally block?

The "finally" block is used to execute the important code of the program. It is executed whether an exception is handled or not. In other words, we can say that finally block is the block which is always executed. Finally block follows try or catch block. If you don't handle the exception, before terminating the program, JVM runs finally block, (if any). The finally block is mainly used to place the cleanup code such as closing a file or closing a connection. Here, we must know that for each try block there can be zero or more catch blocks, but only one finally block. The finally block will not be executed if program exits(either by calling System.exit() or by causing a fatal error that causes the process to abort).

353. Can finally block be used without a catch?

Yes, According to the definition of finally block, it must be followed by a try or catch block, therefore, we can use try block instead of catch.

354. Is there any case when finally will not be executed?

Finally block will not be executed if program exits(either by calling System.exit() or by causing a fatal error that causes the process to abort).

355. What is the difference between throw and throws?

throw keyword

- The throw keyword is used to throw an exception explicitly.
- The checked exceptions cannot be propagated with throw only.
- The throw keyword is followed by an instance.
- The throw keyword is used within the method.
- You cannot throw multiple exceptions.

throws keyword

- The throws keyword is used to declare an exception.
- The checked exception can be propagated with throws
- The throws keyword is followed by class.
- The throws keyword is used with the method signature.

- You can declare multiple exceptions,e.g., public void method()throws IOException, SQLException.

356. Can an exception be rethrown?

Yes.

357. Can subclass overriding method declare an exception if parent class method doesn't throw an exception?

Yes but only unchecked exception not checked.

358. What is String Pool?

String pool is the space reserved in the heap memory that can be used to store the strings. The main advantage of using the String pool is whenever we create a string literal; the JVM checks the "string constant pool" first. If the string already exists in the pool, a reference to the pooled instance is returned. If the string doesn't exist in the pool, a new string instance is created and placed in the pool. Therefore, it saves the memory by avoiding the duplicacy.

359. Why are the objects immutable in java?

Because Java uses the concept of the string literal. Suppose there are five reference variables, all refer to one object "sachin". If one reference variable changes the value of the object, it will be affected by all the reference variables. That is why string objects are immutable in java.

360. Why java uses the concept of the string literal?

To make Java more memory efficient (because no new objects are created if it exists already in the string constant pool).

361. How many objects will be created in the following code?

String s = new String("Welcome");

Two objects, one in string constant pool and other in non-pool(heap).

362. What are the differences between String and StringBuffer?

String

- The String class is immutable.
- The String is slow and consumes more memory when you concat too many strings because every time it creates a new instance.
- The String class overrides the equals() method of Object class. So you can compare the contents of two strings by equals() method.

StringBuffer

- The StringBuffer class is mutable.
- The StringBuffer is fast and consumes less memory when you cancat strings.
- The StringBuffer class doesn't override the equals() method of Object class.

363. What are the differences between StringBuffer and StringBuilder?

StringBuffer

- StringBuffer is synchronized, i.e., thread safe. It means two threads can't call the methods of StringBuffer simultaneously.
- StringBuffer is less efficient than StringBuilder.

StringBuilder

- StringBuilder is non-synchronized,i.e., not thread safe. It means two threads can call the methods of StringBuilder simultaneously.
- StringBuilder is more efficient than StringBuffer.

364. How can we create an immutable class in Java?

We can create an immutable class by defining a final class having all of its members as final.

365. What is the purpose of toString() method in Java?

The toString() method returns the string representation of an object. If you print any object, java compiler internally invokes the toString() method on the object. So overriding the toString() method, returns the desired output, it can be the state of an object, etc. depending upon your implementation. By overriding the toString() method of the Object class, we can return the values of the object, so we don't need to write much code.

366. Why CharArray() is preferred over String to store the password?

String stays in the string pool until the garbage is collected. If we store the password into a string, it stays in the memory for a longer period, and anyone having the memory-dump can extract the password as clear text. On the other hand, Using CharArray allows us to set it to blank whenever we are done with the password. It avoids the security threat with the string by enabling us to control the memory.

367. Name some classes present in java.util.regex package.

There are the following classes and interfaces present in java.util.regex package.

- MatchResult Interface
- Matcher class

- Pattern class
- PatternSyntaxException class

368. How the metacharacters are different from the ordinary characters?

Metacharacters have the special meaning to the regular expression engine. The metacharacters are ^, $, ., *, +, etc. The regular expression engine does not consider them as the regular characters. To enable the regular expression engine treating the metacharacters as ordinary characters, we need to escape the metacharacters with the backslash.

369. Write a regular expression to validate a password. A password must start with an alphabet and followed by alphanumeric characters; Its length must be in between 8 to 20.

The regular expression for the above criteria will be: ^[a-zA-Z][a-zA-Z0-9]{8,19} where ^ represents the start of the regex, [a-zA-Z] represents that the first character must be an alphabet, [a-zA-Z0-9] represents the alphanumeric character, {8,19} represents that the length of the password must be in between 8 and 20.

370. What are the advantages of Java inner classes?

There are two types of advantages of Java inner classes.

- Nested classes represent a special type of relationship that is it can access all the members (data members and methods) of the outer class including private.
- Nested classes are used to develop a more readable and maintainable code because it logically groups classes and interfaces in one place only.
- Code Optimization: It requires less code to write.

371. What is a nested class?

The nested class can be defined as the class which is defined inside another class or interface. We use the nested class to logically group classes and interfaces in one place so that it can be more readable and maintainable. A nested class can access all the data members of the outer class including private data members and methods. The syntax of the nested class is defined below.

```
class Java_Outer_class{
 //code
 class Java_Nested_class{
```

```
 //code
 }
}
```

There are two types of nested classes, static nested class, and non-static nested class. The non-static nested class can also be called as inner-class

372. What are the disadvantages of using inner classes?

There are the following main disadvantages of using inner classes.

- Inner classes increase the total number of classes used by the developer and therefore increases the workload of JVM since it has to perform some routine operations for those extra classes which result in slower performance.
- IDEs provide less support to the inner classes as compare to the top level classes and therefore it annoys the developers while working with inner classes.

373. What are the types of inner classes (non-static nested class) used in Java?

There are mainly three types of inner classes used in Java.

Member Inner Class: A class created within class and outside method.

Anonymous Inner Class: A class created for implementing an interface or extending class. Its name is decided by the java compiler.

Local Inner Class: A class created within the method.

374. Is there any difference between nested classes and inner classes?

Yes, inner classes are non-static nested classes. In other words, we can say that inner classes are the part of nested classes.

375. Can we access the non-final local variable, inside the local inner class?

No, the local variable must be constant if you want to access it in the local inner class.

376. What are anonymous inner classes?

Anonymous inner classes are the classes that are automatically declared and instantiated within an expression. We cannot apply different access modifiers to them. Anonymous class cannot be static, and cannot define any static fields, method, or class. In other words, we can say that it a class without the name and can have only one object that is created by its definition.

377. What is the nested interface?

An Interface that is declared inside the interface or class is known as the nested interface. It is static by default. The nested interfaces are used to group related interfaces so that they can be

easy to maintain. The external interface or class must refer to the nested interface. It can't be accessed directly. The nested interface must be public if it is declared inside the interface but it can have any access modifier if declared within the class.

The syntax of the nested interface is given as follows.

```
interface interface_name{
 ...
 interface nested_interface_name{
  ...
 }
}
```

378. Can a class have an interface?

Yes, an interface can be defined within the class. It is called a nested interface.

379. Can an Interface have a class?

Yes, they are static implicitly.

380. What is Garbage Collection?

Garbage collection is a process of reclaiming the unused runtime objects. It is performed for memory management. In other words, we can say that It is the process of removing unused objects from the memory to free up space and make this space available for Java Virtual Machine. Due to garbage collection java gives 0 as output to a variable whose value is not set, i.e., the variable has been defined but not initialized. For this purpose, we were using free() function in the C language and delete() in C++. In Java, it is performed automatically. So, java provides better memory management.

381. What is gc()?

The gc() method is used to invoke the garbage collector for cleanup processing. This method is found in System and Runtime classes. This function explicitly makes the Java Virtual Machine free up the space occupied by the unused objects so that it can be utilized or reused.

382. How is garbage collection controlled?

Garbage collection is managed by JVM. It is performed when there is not enough space in the memory and memory is running low. We can externally call the System.gc() for the garbage collection. However, it depends upon the JVM whether to perform it or not.

383. How can an object be unreferenced?

There are many ways:

- By nulling the reference
- By assigning a reference to another
- By anonymous object etc.

1) By nulling a reference:

Employee e=new Employee();

e=null;

2) By assigning a reference to another:

Employee e1=new Employee();

Employee e2=new Employee();

e1=e2;//now the first object referred by e1 is available for garbage collection

3) By anonymous object:

new Employee();

384. What is the purpose of the finalize() method?

The finalize() method is invoked just before the object is garbage collected. It is used to perform cleanup processing. The Garbage collector of JVM collects only those objects that are created by new keyword. So if you have created an object without new, you can use the finalize method to perform cleanup processing (destroying remaining objects). The cleanup processing is the process to free up all the resources, network which was previously used and no longer needed. It is essential to remember that it is not a reserved keyword, finalize method is present in the object class hence it is available in every class as object class is the superclass of every class in java. Here, we must note that neither finalization nor garbage collection is guaranteed.

385. Can an unreferenced object be referenced again?

Yes,

386. What kind of thread is the Garbage collector thread?

Daemon thread.

387. What is the difference between final, finally and finalize?

final

- final is used to apply restrictions on class, method, and variable.The final class can't be inherited, final method can't be overridden, and final variable value can't be changed.

- final is a keyword.

finally

- finally is used to place important code, it will be executed whether an exception is handled or not.
- finally is a block

finalize

- finalize is used to perform clean up processing just before an object is garbage collected.
- finalize is a method.

388. What is the purpose of the Runtime class?

Java Runtime class is used to interact with a java runtime environment. Java Runtime class provides methods to execute a process, invoke GC, get total and free memory, etc. There is only one instance of java.lang.Runtime class is available for one java application. The Runtime.getRuntime() method returns the singleton instance of Runtime class.

389. How will you invoke any external process in Java?

By Runtime.getRuntime().exec(?) method.

390. What do you understand by an IO stream?

The stream is a sequence of data that flows from source to destination. It is composed of bytes.

In Java, three streams are created for us automatically.

- System.out: standard output stream
- System.in: standard input stream
- System.err: standard error stream

391. What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy?

The Reader/Writer class hierarchy is character-oriented, and the InputStream/OutputStream class hierarchy is byte-oriented. The ByteStream classes are used to perform input-output of 8-bit bytes whereas the CharacterStream classes are used to perform the input/output for the 16-bit Unicode system. There are many classes in the ByteStream class hierarchy, but the most frequently used classes are FileInputStream and FileOutputStream. The most frequently used classes CharacterStream class hierarchy is FileReader and FileWriter.

392. What are the super most classes for all the streams?

All the stream classes can be divided into two types of classes that are ByteStream classes and CharacterStream Classes. The ByteStream classes are further divided into InputStream classes and OutputStream classes. CharacterStream classes are also divided into Reader classes and Writer classes. The SuperMost classes for all the InputStream classes is java.io.InputStream and for all the output stream classes is java.io.OutPutStream. Similarly, for all the reader classes, the super-most class is java.io.Reader, and for all the writer classes, it is java.io.Writer.

393. What are the FileInputStream and FileOutputStream?

Java FileOutputStream is an output stream used for writing data to a file. If you have some primitive values to write into a file, use FileOutputStream class. You can write byte-oriented as well as character-oriented data through the FileOutputStream class. However, for character-oriented data, it is preferred to use FileWriter than FileOutputStream

Java FileInputStream class obtains input bytes from a file. It is used for reading byte-oriented data (streams of raw bytes) such as image data, audio, video, etc. You can also read character-stream data. However, for reading streams of characters, it is recommended to use FileReader class.

394. What is the purpose of using BufferedInputStream and BufferedOutputStream classes?

Java BufferedOutputStream class is used for buffering an output stream. It internally uses a buffer to store data. It adds more efficiency than to write data directly into a stream. So, it makes the performance fast. Whereas, Java BufferedInputStream class is used to read information from the stream. It internally uses the buffer mechanism to make the performance fast.

395. How to set the Permissions to a file in Java?

In Java, FilePermission class is used to alter the permissions set on a file. Java FilePermission class contains the permission related to a directory or file. All the permissions are related to the path. The path can be of two types:

D:\\IO\\-: It indicates that the permission is associated with all subdirectories and files recursively.

D:\\IO*: It indicates that the permission is associated with all directory and files within this directory excluding subdirectories.

396. What are FilterStreams?

FilterStream classes are used to add additional functionalities to the other stream classes. FilterStream classes act like an interface which read the data from a stream, filters it, and pass the filtered data to the caller. The FilterStream classes provide extra functionalities like adding line numbers to the destination file, etc.

397. What is an I/O filter?

An I/O filter is an object that reads from one stream and writes to another, usually altering the data in some way as it is passed from one stream to another. Many Filter classes that allow a user to make a chain using multiple input streams. It generates a combined effect on several filters.

398. In Java, How many ways you can take input from the console?

In Java, there are three ways by using which, we can take input from the console.

Using BufferedReader class: we can take input from the console by wrapping System.in into an InputStreamReader and passing it into the BufferedReader. It provides an efficient reading as the input gets buffered.

Using Scanner class: The Java Scanner class breaks the input into tokens using a delimiter that is whitespace by default. It provides many methods to read and parse various primitive values. Java Scanner class is widely used to parse text for string and primitive types using a regular expression. Java Scanner class extends Object class and implements Iterator and Closeable interfaces.

Using Console class: The Java Console class is used to get input from the console. It provides methods to read texts and passwords. If you read the password using the Console class, it will not be displayed to the user. The java.io.Console class is attached to the system console internally.

399. What is serialization?

Serialization in Java is a mechanism of writing the state of an object into a byte stream. It is used primarily in Hibernate, RMI, JPA, EJB and JMS technologies. It is mainly used to travel object's state on the network (which is known as marshaling). Serializable interface is used to perform serialization. It is helpful when you require to save the state of a program to storage such as the file. At a later point of time, the content of this file can be restored using deserialization. It is also required to implement RMI(Remote Method Invocation). With the help of RMI, it is possible to invoke the method of a Java object on one machine to another machine.

400. How can you make a class serializable in Java?

A class can become serializable by implementing the Serializable interface.

401. How can you avoid serialization in child class if the base class is implementing the Serializable interface?

It is very tricky to prevent serialization of child class if the base class is intended to implement the Serializable interface. However, we cannot do it directly, but the serialization can be avoided by implementing the writeObject() or readObject() methods in the subclass and throw NotSerializableException from these methods.

402. Can a Serialized object be transferred via network?

Yes, we can transfer a serialized object via network because the serialized object is stored in the memory in the form of bytes and can be transmitted over the network. We can also write the serialized object to the disk or the database.

403. What is Deserialization?

Deserialization is the process of reconstructing the object from the serialized state. It is the reverse operation of serialization. An ObjectInputStream deserializes objects and primitive data written using an ObjectOutputStream.

```
import java.io.*;
class Depersist{
 public static void main(String args[])throws Exception{
 ObjectInputStream in=new ObjectInputStream(new FileInputStream("f.txt"));
  Student s=(Student)in.readObject();
  System.out.println(s.id+" "+s.name);
    in.close();
 }
}
```

output

211 ravi

404. What is the transient keyword?

If you define any data member as transient, it will not be serialized. By determining transient keyword, the value of variable need not persist when it is restored.

405. What is Externalizable?

The Externalizable interface is used to write the state of an object into a byte stream in a compressed format. It is not a marker interface.

406. What is the difference between Serializable and Externalizable interface?

Serializable

- The Serializable interface does not have any method, i.e., it is a marker interface.
- It is used to "mark" Java classes so that objects of these classes may get the certain capability.
- It is easy to implement but has the higher performance cost.
- No class constructor is called in serialization.

Externalizable

- The Externalizable interface contains is not a marker interface, It contains two methods, i.e., writeExternal() and readExternal().
- The Externalizable interface provides control of the serialization logic to the programmer.
- It is used to perform the serialization and often result in better performance.
- We must call a public default constructor while using this interface.

407. Give a brief description of Java socket programming?

Java Socket programming is used for communication between the applications running on different JRE. Java Socket programming can be connection-oriented or connectionless. Socket and ServerSocket classes are used for connection-oriented socket programming and DatagramSocket, and DatagramPacket classes are used for connectionless socket programming. The client in socket programming must know two information:

- IP address of the server
- port number

408. What is Socket?

A socket is simply an endpoint for communications between the machines. It provides the connection mechanism to connect the two computers using TCP. The Socket class can be used to create a socket.

409. What are the steps that are followed when two computers connect through TCP?

There are the following steps that are performed when two computers connect through TCP.

- The ServerSocket object is instantiated by the server which denotes the port number to which, the connection will be made.
- After instantiating the ServerSocket object, the server invokes accept() method of ServerSocket class which makes server wait until the client attempts to connect to the server on the given port.
- Meanwhile, the server is waiting, a socket is created by the client by instantiating Socket class. The socket class constructor accepts the server port number and server name.
- The Socket class constructor attempts to connect with the server on the specified name. If the connection is established, the client will have a socket object that can communicate with the server.
- The accept() method invoked by the server returns a reference to the new socket on the server that is connected with the server.

410. How do I convert a numeric IP address like 192.18.97.39 into a hostname like java.sun.com?

By InetAddress.getByName("192.18.97.39").getHostName() where 192.18.97.39 is the IP address.

411. What is the reflection?

Reflection is the process of examining or modifying the runtime behavior of a class at runtime. The java.lang.Class class provides various methods that can be used to get metadata, examine and change the runtime behavior of a class. The java.lang and java.lang.reflect packages provide classes for java reflection. It is used in:

- IDE (Integrated Development Environment), e.g., Eclipse, MyEclipse, NetBeans.
- Debugger
- Test Tools, etc.

412. What is the purpose of using java.lang.Class class?

The java.lang.Class class performs mainly two tasks:

- Provides methods to get the metadata of a class at runtime.
- Provides methods to examine and change the runtime behavior of a class.

413. What are the ways to instantiate the Class class?

There are three ways to instantiate the Class class.

- forName() method of Class class: The forName() method is used to load the class dynamically. It returns the instance of Class class. It should be used if you know the fully qualified name of the class. This cannot be used for primitive types.
- getClass() method of Object class: It returns the instance of Class class. It should be used if you know the type. Moreover, it can be used with primitives.
- the .class syntax: If a type is available, but there is no instance then it is possible to obtain a Class by appending ".class" to the name of the type. It can be used for primitive data type also.

414. What is the purpose of using javap?

The javap command disassembles a class file. The javap command displays information about the fields, constructors and methods present in a class file.

Syntax

javap fully_class_name

415. Can you access the private method from outside the class?

Yes, by changing the runtime behavior of a class if the class is not secured.

416. What are wrapper classes?

Wrapper classes are classes that allow primitive types to be accessed as objects. In other words, we can say that wrapper classes are built-in java classes which allow the conversion of objects to primitives and primitives to objects. The process of converting primitives to objects is called autoboxing, and the process of converting objects to primitives is called unboxing. There are eight wrapper classes present in java.lang package is given below.

Primitive Type	Wrapper class
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer

long Long

float Float

double Double

417. What are autoboxing and unboxing? When does it occur?

The autoboxing is the process of converting primitive data type to the corresponding wrapper class object, eg., int to Integer. The unboxing is the process of converting wrapper class object to primitive data type. For eg., integer to int. Unboxing and autoboxing occur automatically in Java. However, we can externally convert one into another by using the methods like valueOf() or xxxValue().

It can occur whenever a wrapper class object is expected, and primitive data type is provided or vice versa.

- Adding primitive types into Collection like ArrayList in Java.
- Creating an instance of parameterized classes ,e.g., ThreadLocal which expect Type.
- Java automatically converts primitive to object whenever one is required and another is provided in the method calling.
- When a primitive type is assigned to an object type.

418. What is object cloning?

The object cloning is a way to create an exact copy of an object. The clone() method of the Object class is used to clone an object. The java.lang.Cloneable interface must be implemented by the class whose object clone we want to create. If we don't implement Cloneable interface, clone() method generates CloneNotSupportedException. The clone() method is defined in the Object class. The syntax of the clone() method is as follows:

protected Object clone() throws CloneNotSupportedException

419. What are the advantages and disadvantages of object cloning?

Advantage of Object Cloning

You don't need to write lengthy and repetitive codes. Just use an abstract class with a 4- or 5-line long clone() method.

It is the easiest and most efficient way of copying objects, especially if we are applying it to an already developed or an old project. Just define a parent class, implement Cloneable in it, provide the definition of the clone() method and the task will be done.

Clone() is the fastest way to copy the array.

Disadvantage of Object Cloning

To use the Object.clone() method, we have to change many syntaxes to our code, like implementing a Cloneable interface, defining the clone() method and handling CloneNotSupportedException, and finally, calling Object.clone(), etc.

We have to implement the Cloneable interface while it does not have any methods in it. We have to use it to tell the JVM that we can perform a clone() on our object.

Object.clone() is protected, so we have to provide our own clone() and indirectly call Object.clone() from it.

Object.clone() does not invoke any constructor, so we do not have any control over object construction.

If you want to write a clone method in a child class, then all of its superclasses should define the clone() method in them or inherit it from another parent class. Otherwise, the super.clone() chain will fail.

Object.clone() supports only shallow copying, but we will need to override it if we need deep cloning.

420. What is a native method?

A native method is a method that is implemented in a language other than Java. Natives methods are sometimes also referred to as foreign methods.

421. What is the purpose of the strictfp keyword?

Java strictfp keyword ensures that you will get the same result on every platform if you perform operations in the floating-point variable. The precision may differ from platform to platform that is why java programming language has provided the strictfp keyword so that you get the same result on every platform. So, now you have better control over the floating-point arithmetic.

422. What is the purpose of the System class?

The purpose of the System class is to provide access to system resources such as standard input and output. It cannot be instantiated. Facilities provided by System class are given below.

Standard input

Error output streams

Standard output

utility method to copy the portion of an array

utilities to load files and libraries

There are the three fields of Java System class, i.e., static printstream err, static inputstream in, and standard output stream.

423. What comes to mind when someone mentions a shallow copy in Java?

Object cloning.

424. What is a singleton class?

Singleton class is the class which can not be instantiated more than once. To make a class singleton, we either make its constructor private or use the static getInstance method.

425. Which containers use a border layout as their default layout?

The Window, Frame and Dialog classes use a border layout as their default layout.

426. Which containers use a FlowLayout as their default layout?

The Panel and Applet classes use the FlowLayout as their default layout.

427. What are peerless components?

The lightweight component of Swing is called peerless components. Spring has its libraries, so it does not use resources from the Operating System, and hence it has lightweight components.

428. is there is any difference between a Scrollbar and a ScrollPane?

The Scrollbar is a Component whereas the ScrollPane is a Container. A ScrollPane handles its events and performs its scrolling.

429. What is a lightweight component?

Lightweight components are the one which does not go with the native call to obtain the graphical units. They share their parent component graphical units to render them. For example, Swing components, and JavaFX Components.

430. What is a heavyweight component?

The portable elements provided by the operating system are called heavyweight components. AWT is limited to the graphical classes provided by the operating system and therefore, It implements only the minimal subset of screen elements supported by all platforms. The Operating system dependent UI discovery tools are called heavyweight components.

431. What is an applet?

An applet is a small java program that runs inside the browser and generates dynamic content. It is embedded in the webpage and runs on the client side. It is secured and takes less response time. It can be executed by browsers running under many platforms, including Linux, Windows, Mac Os, etc. However, the plugins are required at the client browser to execute the applet.

When an applet is created, the following methods are invoked in order.

init()

start()

paint()

When an applet is destroyed, the following functions are invoked in order.

stop()

destroy()

432. Can you write a Java class that could be used both as an applet as well as an application?

Yes. Add a main() method to the applet.

433. **What is Locale?**

A Locale object represents a specific geographical, political, or cultural region. This object can be used to get the locale-specific information such as country name, language, variant, etc.

```
import java.util.*;
public class LocaleExample {
public static void main(String[] args) {
Locale locale=Locale.getDefault();
//Locale locale=new Locale("fr","fr");//for the specific locale
System.out.println(locale.getDisplayCountry());
System.out.println(locale.getDisplayLanguage());
System.out.println(locale.getDisplayName());
System.out.println(locale.getISO3Country());
System.out.println(locale.getISO3Language());
System.out.println(locale.getLanguage());
System.out.println(locale.getCountry());

}
```

```
}
```

Output:

United States

English

English (United States)

USA

eng

en

US

434. How will you load a specific locale?

By ResourceBundle.getBundle(?) method.

435. What is a JavaBean?

JavaBean is a reusable software component written in the Java programming language, designed to be manipulated visually by a software development environment, like JBuilder or VisualAge for Java. t. A JavaBean encapsulates many objects into one object so that we can access this object from multiple places. Moreover, it provides the easy maintenance.

436. What is the purpose of using the Java bean?

According to Java white paper, it is a reusable software component. A bean encapsulates many objects into one object so that we can access this object from multiple places. Moreover, it provides the easy maintenance.

437. What do you understand by the bean persistent property?

The persistence property of Java bean comes into the act when the properties, fields, and state information are saved to or retrieve from the storage.

438. What is RMI?

The RMI (Remote Method Invocation) is an API that provides a mechanism to create the distributed application in java. The RMI allows an object to invoke methods on an object running in another JVM. The RMI provides remote communication between the applications using two objects stub and skeleton.

439. What is the purpose of stub and skeleton?

Stub

The stub is an object, acts as a gateway for the client side. All the outgoing requests are routed through it. It resides at the client side and represents the remote object. When the caller invokes the method on the stub object, it does the following tasks:

- It initiates a connection with remote Virtual Machine (JVM).
- It writes and transmits (marshals) the parameters to the remote Virtual Machine (JVM).
- It waits for the result.
- It reads (unmarshals) the return value or exception.
- It finally, returns the value to the caller.

Skeleton

The skeleton is an object, acts as a gateway for the server side object. All the incoming requests are routed through it. When the skeleton receives the incoming request, it does the following tasks:

- It reads the parameter for the remote method.
- It invokes the method on the actual remote object.
- It writes and transmits (marshals) the result to the caller.

440. 439) What are the steps involved to write RMI based programs?

There are 6 steps which are performed to write RMI based programs.

- Create the remote interface.
- Provide the implementation of the remote interface.
- Compile the implementation class and create the stub and skeleton objects using the rmic tool.
- Start the registry service by the rmiregistry tool.
- Create and start the remote application.
- Create and start the client application.

441. What is the use of HTTP-tunneling in RMI?

HTTP tunneling can be defined as the method which doesn't need any setup to work within the firewall environment. It handles the HTTP connections through the proxy servers. However, it does not allow outbound TCP connections.

442. What is JRMP?

JRMP (Java Remote Method Protocol) can be defined as the Java-specific, stream-based protocol which looks up and refers to the remote objects. It requires both client and server to use Java objects. It is wire level protocol which runs under RMI and over TCP/IP.

443. Can RMI and CORBA based applications interact?

Yes, they can. RMI is available with IIOP as the transport protocol instead of JRMP.

444. What is multithreading?

Multithreading is a process of executing multiple threads simultaneously. Multithreading is used to obtain the multitasking. It consumes less memory and gives the fast and efficient performance. Its main advantages are:

- Threads share the same address space.
- The thread is lightweight.
- The cost of communication between the processes is low.

445. What is the thread?

A thread is a lightweight subprocess. It is a separate path of execution because each thread runs in a different stack frame. A process may contain multiple threads. Threads share the process resources, but still, they execute independently.

446. Differentiate between process and thread?

There are the following differences between the process and thread.

A Program in the execution is called the process whereas; A thread is a subset of the process

Processes are independent whereas threads are the subset of process.

Process have different address space in memory, while threads contain a shared address space.

Context switching can be faster between the threads as compared to context switching between the threads.

Inter-process communication is slower and expensive than inter-thread communication.

Any change in Parent process doesn't affect the child process whereas changes in parent thread can affect the child thread.

447. What do you understand by inter-thread communication?

- The process of communication between synchronized threads is termed as inter-thread communication.
- Inter-thread communication is used to avoid thread polling in Java.

- The thread is paused running in its critical section, and another thread is allowed to enter (or lock) in the same critical section to be executed.
- It can be obtained by wait(), notify(), and notifyAll() methods.

448. What is the purpose of wait() method in Java?

The wait() method is provided by the Object class in Java. This method is used for inter-thread communication in Java. The java.lang.Object.wait() is used to pause the current thread, and wait until another thread does not call the notify() or notifyAll() method. Its syntax is given below.

public final void wait()

449. Why must wait() method be called from the synchronized block?

We must call the wait method otherwise it will throw **java.lang.IllegalMonitorStateException** exception. Moreover, we need wait() method for inter-thread communication with notify() and notifyAll(). Therefore It must be present in the synchronized block for the proper and correct communication.

450. What are the advantages of multithreading?

Multithreading programming has the following advantages:

- Multithreading allows an application/program to be always reactive for input, even already running with some background tasks
- Multithreading allows the faster execution of tasks, as threads execute independently.
- Multithreading provides better utilization of cache memory as threads share the common memory resources.
- Multithreading reduces the number of the required server as one server can execute multiple threads at a time.

451. What are the states in the lifecycle of a Thread?

A thread can have one of the following states during its lifetime:

1. New: In this state, a Thread class object is created using a new operator, but the thread is not alive. Thread doesn't start until we call the start() method.
2. Runnable: In this state, the thread is ready to run after calling the start() method. However, the thread is not yet selected by the thread scheduler.
3. Running: In this state, the thread scheduler picks the thread from the ready state, and the thread is running.

4. Waiting/Blocked: In this state, a thread is not running but still alive, or it is waiting for the other thread to finish.

5. Dead/Terminated: A thread is in terminated or dead state when the run() method exits.

452. What is the difference between preemptive scheduling and time slicing?

Under preemptive scheduling, the highest priority task executes until it enters the waiting or dead states or a higher priority task comes into existence. Under time slicing, a task executes for a predefined slice of time and then reenters the pool of ready tasks. The scheduler then determines which task should execute next, based on priority and other factors.

453. What is context switching?

In Context switching the state of the process (or thread) is stored so that it can be restored and execution can be resumed from the same point later. Context switching enables the multiple processes to share the same CPU.

454. What does join() method?

The join() method waits for a thread to die. In other words, it causes the currently running threads to stop executing until the thread it joins with completes its task. Join method is overloaded in Thread class in the following ways.

- public void join()throws InterruptedException
- public void join(long milliseconds)throws InterruptedException

455. Describe the purpose and working of sleep() method.

The sleep() method in java is used to block a thread for a particular time, which means it pause the execution of a thread for a specific time. There are two methods of doing so.

Syntax:

public static void sleep(long milliseconds)throws InterruptedException

public static void sleep(long milliseconds, int nanos)throws InterruptedException

Working of sleep() method

When we call the sleep() method, it pauses the execution of the current thread for the given time and gives priority to another thread(if available). Moreover, when the waiting time completed then again previous thread changes its state from waiting to runnable and comes in running state, and the whole process works so on till the execution doesn't complete.

456. What is the difference between wait() and sleep() method?

wait()

The wait() method is defined in Object class.

The wait() method releases the lock.

sleep()

The sleep() method is defined in Thread class.

The sleep() method doesn't release the lock.

457. Is it possible to start a thread twice?

No, we cannot restart the thread, as once a thread started and executed, it goes to the Dead state. Therefore, if we try to start a thread twice, it will give a runtimeException "java.lang.IllegalThreadStateException".

458. Can we call the run() method instead of start()?

Yes, calling run() method directly is valid, but it will not work as a thread instead it will work as a normal object. There will not be context-switching between the threads. When we call the start() method, it internally calls the run() method, which creates a new stack for a thread while directly calling the run() will not create a new stack.

459. What about the daemon threads?

The daemon threads are the low priority threads that provide the background support and services to the user threads. Daemon thread gets automatically terminated by the JVM if the program remains with the daemon thread only, and all other user threads are ended/died. There are two methods for daemon thread available in the Thread class:

- public void setDaemon(boolean status): It used to mark the thread daemon thread or a user thread.
- public boolean isDaemon(): It checks the thread is daemon or not.

460. When should we interrupt a thread?

We should interrupt a thread when we want to break out the sleep or wait state of a thread. We can interrupt a thread by calling the interrupt() throwing the InterruptedException.

461. What is the synchronization?

Synchronization is the capability to control the access of multiple threads to any shared resource. It is used:

1. To prevent thread interference.
2. To prevent consistency problem.

When the multiple threads try to do the same task, there is a possibility of an erroneous result, hence to remove this issue, Java uses the process of synchronization which allows only one thread to be executed at a time. Synchronization can be achieved in three ways:

by the synchronized method

by synchronized block

by static synchronization

Syntax for synchronized block

```
synchronized(object reference expression)
  {
    //code block
  }
```

462. What is the purpose of the Synchronized block?

The Synchronized block can be used to perform synchronization on any specific resource of the method. Only one thread at a time can execute on a particular resource, and all other threads which attempt to enter the synchronized block are blocked.

- Synchronized block is used to lock an object for any shared resource.
- The scope of the synchronized block is limited to the block on which, it is applied. Its scope is smaller than a method.

463. Can Java object be locked down for exclusive use by a given thread?

Yes. You can lock an object by putting it in a "synchronized" block. The locked object is inaccessible to any thread other than the one that explicitly claimed it.

464. What is static synchronization?

If you make any static method as synchronized, the lock will be on the class not on the object. If we use the synchronized keyword before a method so it will lock the object (one thread can access an object at a time) but if we use static synchronized so it will lock a class (one thread can access a class at a time).

465. What is the difference between notify() and notifyAll()?

The notify() is used to unblock one waiting thread whereas notifyAll() method is used to unblock all the threads in waiting state.

466. What is the deadlock?

Deadlock is a situation in which every thread is waiting for a resource which is held by some other waiting thread. In this situation, Neither of the thread executes nor it gets the chance to be executed. Instead, there exists a universal waiting state among all the threads. Deadlock is a very complicated situation which can break our code at runtime.

467. How to detect a deadlock condition? How can it be avoided?

We can detect the deadlock condition by running the code on cmd and collecting the Thread Dump, and if any deadlock is present in the code, then a message will appear on cmd.

Ways to avoid the deadlock condition in Java:

Avoid Nested lock: Nested lock is the common reason for deadlock as deadlock occurs when we provide locks to various threads so we should give one lock to only one thread at some particular time.

Avoid unnecessary locks: we must avoid the locks which are not required.

Using thread join: Thread join helps to wait for a thread until another thread doesn't finish its execution so we can avoid deadlock by maximum use of join method.

468. What is Thread Scheduler in java?

In Java, when we create the threads, they are supervised with the help of a Thread Scheduler, which is the part of JVM. Thread scheduler is only responsible for deciding which thread should be executed. Thread scheduler uses two mechanisms for scheduling the threads: Preemptive and Time Slicing.

Java thread scheduler also works for deciding the following for a thread:

- It selects the priority of the thread.
- It determines the waiting time for a thread
- It checks the Nature of thread

469. Does each thread have its stack in multithreaded programming?

Yes, in multithreaded programming every thread maintains its own or separate stack area in memory due to which every thread is independent of each other.

470. How is the safety of a thread achieved?

If a method or class object can be used by multiple threads at a time without any race condition, then the class is thread-safe. Thread safety is used to make a program safe to use in multithreaded programming. It can be achieved by the following ways:

- Synchronization
- Using Volatile keyword
- Using a lock based mechanism
- Use of atomic wrapper classes

471. What is race-condition?

A Race condition is a problem which occurs in the multithreaded programming when various threads execute simultaneously accessing a shared resource at the same time. The proper use of synchronization can avoid the Race condition.

472. What is the volatile keyword in java?

Volatile keyword is used in multithreaded programming to achieve the thread safety, as a change in one volatile variable is visible to all other threads so one variable can be used by one thread at a time.

473. What do you understand by thread pool?

Java Thread pool represents a group of worker threads, which are waiting for the task to be allocated.

Threads in the thread pool are supervised by the service provider which pulls one thread from the pool and assign a job to it.

After completion of the given task, thread again came to the thread pool.

The size of the thread pool depends on the total number of threads kept at reserve for execution.

The advantages of the thread pool are :

Using a thread pool, performance can be enhanced.

Using a thread pool, better system stability can occur.

474. What is BlockingQueue?

The java.util.concurrent.BlockingQueue is the subinterface of Queue that supports the operations such as waiting for the space availability before inserting a new value or waiting for the queue to become non-empty before retrieving an element from it.

475. What is the Collection framework in Java?

Collection Framework is a combination of classes and interface, which is used to store and manipulate the data in the form of objects. It provides various classes such as ArrayList, Vector, Stack, and HashSet, etc. and interfaces such as List, Queue, Set, etc. for this purpose.

476. What are the main differences between array and collection?

Array and Collection are somewhat similar regarding storing the references of objects and manipulating the data, but they differ in many ways. The main differences between the array and Collection are defined below:

- Arrays are always of fixed size, i.e., a user can not increase or decrease the length of the array according to their requirement or at runtime, but In Collection, size can be changed dynamically as per need.
- Arrays can only store homogeneous or similar type objects, but in Collection, heterogeneous objects can be stored.
- Arrays cannot provide the ?ready-made? methods for user requirements as sorting, searching, etc. but Collection includes readymade methods to use.

477. Explain various interfaces used in Collection framework?

Collection framework implements various interfaces, Collection interface and Map interface (java.util.Map) are the mainly used interfaces of Java Collection Framework. List of interfaces of Collection Framework is given below:

1. Collection interface: Collection (java.util.Collection) is the primary interface, and every collection must implement this interface.

Syntax:

public interface Collection<E>extends Iterable

Where <E> represents that this interface is of Generic type

2. List interface: List interface extends the Collection interface, and it is an ordered collection of objects. It contains duplicate elements. It also allows random access of elements.

Syntax:

public interface List<E> extends Collection<E>

3. Set interface: Set (java.util.Set) interface is a collection which cannot contain duplicate elements. It can only include inherited methods of Collection interface

Syntax:

public interface Set<E> extends Collection<E>

Queue interface: Queue (java.util.Queue) interface defines queue data structure, which stores the elements in the form FIFO (first in first out).

Syntax:

public interface Queue<E> extends Collection<E>

4. Dequeue interface: it is a double-ended-queue. It allows the insertion and removal of elements from both ends. It implants the properties of both Stack and queue so it can perform LIFO (Last in first out) stack and FIFO (first in first out) queue, operations.

Syntax:

public interface Dequeue<E> extends Queue<E>

5. Map interface: A Map (java.util.Map) represents a key, value pair storage of elements. Map interface does not implement the Collection interface. It can only contain a unique key but can have duplicate elements. There are two interfaces which implement Map in java that are Map interface and Sorted Map.

478. What is the difference between ArrayList and Vector?

ArrayList

- ArrayList is not synchronized.
- ArrayList is not a legacy class
- ArrayList increases its size by 50% of the array size

Vector

- Vector is synchronized.
- Vector is a legacy class.
- Vector increases its size by doubling the array size.

479. What is the difference between ArrayList and LinkedList?

ArrayList

- ArrayList uses a dynamic array.
- ArrayList is not efficient for manipulation because too much is required.
- ArrayList is better to store and fetch data.
- ArrayList provides random access.
- ArrayList takes less memory overhead as it stores only object

LinkedList

- LinkedList uses a doubly linked list.
- LinkedList is efficient for manipulation.
- LinkedList is better to manipulate data.
- LinkedList does not provide random access.
- LinkedList takes more memory overhead, as it stores the object as well as the address of that object.

480. What is the difference between Iterator and Enumeration?

Iterator

The Iterator can traverse legacy and non-legacy elements

The Iterator is fail-fast.

The Iterator is slower than Enumeration.

The Iterator can perform remove operation while traversing the collection

Enumeration

Enumeration can traverse only legacy elements.

Enumeration is not fail-fast.

Enumeration is faster than Iterator.

The Enumeration can perform only traverse operation on the collection.

481. What is the difference between List and Set?

The List and Set both extend the collection interface. However, there are some differences between the both which are listed below.

- The List can contain duplicate elements whereas Set includes unique items.
- The List is an ordered collection which maintains the insertion order whereas Set is an unordered collection which does not preserve the insertion order.
- The List interface contains a single legacy class which is Vector class whereas Set interface does not have any legacy class.
- The List interface can allow n number of null values whereas Set interface only allows a single null value.

482. What is the difference between HashSet and TreeSet?

The HashSet and TreeSet, both classes, implement Set interface. The differences between the both are listed below.

- HashSet maintains no order whereas TreeSet maintains ascending order.
- HashSet impended by hash table whereas TreeSet implemented by a Tree structure.
- HashSet performs faster than TreeSet.
- HashSet is backed by HashMap whereas TreeSet is backed by TreeMap

483. What is the difference between Set and Map?

The differences between the Set and Map are given below.

- Set contains values only whereas Map contains key and values both.
- Set contains unique values whereas Map can contain unique Keys with duplicate values.
- Set holds a single number of null value whereas Map can include a single null key with n number of null values.

484. What is the difference between HashSet and HashMap?

The differences between the HashSet and HashMap are listed below.

- HashSet contains only values whereas HashMap includes the entry (key, value). HashSet can be iterated, but HashMap needs to convert into Set to be iterated.
- HashSet implements Set interface whereas HashMap implements the Map interface
- HashSet cannot have any duplicate value whereas HashMap can contain duplicate values with unique keys.
- HashSet contains the only single number of null value whereas HashMap can hold a single null key with n number of null values.

485. What does the hashCode() method?

The hashCode() method returns a hash code value (an integer number).

The hashCode() method returns the same integer number if two keys (by calling equals() method) are identical.

However, it is possible that two hash code numbers can have different or the same keys.

If two objects do not produce an equal result by using the equals() method, then the hashcode() method will provide the different integer result for both the objects.

486. Why we override equals() method?

The equals method is used to check whether two objects are the same or not. It needs to be overridden if we want to check the objects based on the property.

For example, Employee is a class that has 3 data members: id, name, and salary. However, we want to check the equality of employee object by the salary. Then, we need to override the equals() method.

487. How to synchronize List, Set and Map elements?

Yes, Collections class provides methods to make List, Set or Map elements as synchronized:

public static List synchronizedList(List l){}

public static Set synchronizedSet(Set s){}

public static SortedSet synchronizedSortedSet(SortedSet s){}

public static Map synchronizedMap(Map m){}

public static SortedMap synchronizedSortedMap(SortedMap m){}

488. What is the advantage of the generic collection?

There are three main advantages of using the generic collection.

- If we use the generic class, we don't need typecasting.
- It is type-safe and checked at compile time.
- Generic confirms the stability of the code by making it bug detectable at compile time.

489. What is hash-collision in Hashtable and how it is handled in Java?

Two different keys with the same hash value are known as hash-collision. Two separate entries will be kept in a single hash bucket to avoid the collision. There are two ways to avoid hash-collision.

- Separate Chaining
- Open Addressing

490. What is the Dictionary class?

The Dictionary class provides the capability to store key-value pairs.

491. What is the default size of load factor in hashing based collection?

The default size of load factor is 0.75. The default capacity is computed as initial capacity * load factor. For example, 16 * 0.75 = 12. So, 12 is the default capacity of Map.

492. What do you understand by fail-fast?

The Iterator in java which immediately throws ConcurrentmodificationException, if any structural modification occurs in, is called as a Fail-fast iterator. Fail-fats iterator does not require any extra space in memory.

493. How to synchronize ArrayList?

We can synchronize ArrayList in two ways.

- Using Collections.synchronizedList() method
- Using CopyOnWriteArrayList<T>

494. How to remove duplicates from ArrayList?

There are two ways to remove duplicates from the ArrayList.

- Using HashSet: By using HashSet we can remove the duplicate element from the ArrayList, but it will not then preserve the insertion order.
- Using LinkedHashSet: We can also maintain the insertion order by using LinkedHashSet instead of HashSet.

The Process to remove duplicate elements from ArrayList using the LinkedHashSet:

- Copy all the elements of ArrayList to LinkedHashSet.
- Empty the ArrayList using clear() method, which will remove all the elements from the list.
- Now copy all the elements of LinkedHashset to ArrayList.

495. How to convert ArrayList to Array and Array to ArrayList?

We can convert an Array to ArrayList by using the asList() method of Arrays class. asList() method is the static method of Arrays class and accepts the List object. Consider the following syntax:

Arrays.asList(item)

We can convert an ArrayList to Array using toArray() method of the ArrayList class. Consider the following syntax to convert the ArrayList to the List object.

List_object.toArray(new String[List_object.size()])

496. What is the difference between Array and ArrayList?

The main differences between the Array and ArrayList are given below.

Array

The Array is of fixed size, means we cannot resize the array as per need.

Arrays are of the static type.

Arrays can store primitive data types as well as objects.

ArrayList

ArrayList is not of the fixed size we can change the size dynamically.

ArrayList is of dynamic size.

ArrayList cannot store the primitive data types it can only store the objects.

497. When to use ArrayList and LinkedList?

LinkedLists are better to use for the update operations whereas ArrayLists are better to use for the search operations.

498. What is the difference between Comparable and Comparator?

Comparable

- Comparable provides only one sort of sequence.
- It provides one method named compareTo().
- It is found in java.lang package.
- If we implement the Comparable interface, The actual class is modified.

Comparator

- The Comparator provides multiple sorts of sequences.
- It provides one method named compare().
- It is located in java.util package.
- The actual class is not changed.

499. What is the output of the following Java program?

```
class Test
{
   public static void main (String args[])
   {
      System.out.println(10 + 20 + "Javatpoint");
      System.out.println("Javatpoint" + 10 + 20);
   }
}
```

The output of the above code will be

30Javatpoint

Javatpoint1020

500. What is the output of the following Java program?

```
class Test
{
```

```
   public static void main (String args[])
   {
      System.out.println(10 * 20 + "Javatpoint");
      System.out.println("Javatpoint" + 10 * 20);
   }
}
```

The output of the above code will be

200Javatpoint

Javatpoint200

501. Write a Java program that prints all the values given at command-line.

Program

```
class A{
public static void main(String args[]){
for(int i=0;i<args.length;i++)
System.out.println(args[i]);

}
}
```

compile by > javac A.java

run by > java A sonoo jaiswal 1 3 abc

Output

sonoo

jaiswal

1

3

abc

502. What is the output of the below Java program?

```
public class Test1
{
  public static void main(String[] args) {
```

```
  Integer i = new Integer(201);
  Integer j = new Integer(201);
  if(i == j)
  {
    System.out.println("hello");
  }
  else
  {
    System.out.println("bye");
  }
  }
}
```

Output

Bye

503. How many class files are created on compiling the OuterClass in the following program?

```
public class Person {
String name, age, address;
class Employee{
  float salary=10000;
}
class BusinessMen{
  final String gstin="£4433drt3$";
}
public static void main (String args[])
{
  Person p = new Person();
}
}
```

3 class-files will be created named as Person.class, Person$BusinessMen.class, and Person$Employee.class.

504. What is the output of the following Java program?

```
import java.util.regex.*;
class RegexExample2{
public static void main(String args[]){
System.out.println(Pattern.matches(".s", "as")); //line 4
System.out.println(Pattern.matches(".s", "mk")); //line 5
System.out.println(Pattern.matches(".s", "mst")); //line 6
System.out.println(Pattern.matches(".s", "amms")); //line 7
System.out.println(Pattern.matches("..s", "mas")); //line 8
}}
```

Output

true

false

false

false

true

505. What is the output of the following Java program?

```
public class Test
{
   public static void main (String args[])
   {
      String s1 = "Sharma is a good player";
      String s2 = new String("Sharma is a good player");
      s2 = s2.intern();
      System.out.println(s1 ==s2);
   }
}
```

Output

True

506. What is the output of the following Java program?

```
class Calculation extends Exception
{
   public Calculation()
   {
      System.out.println("Calculation class is instantiated");
   }
   public void add(int a, int b)
   {
      System.out.println("The sum is "+(a+b));
   }
}
public class Main{
    public static void main(String []args){
       try
       {
          throw new Calculation();
       }
       catch(Calculation c){
          c.add(10,20);
       }
   }
}
```

Output

Calculation class is instantiated

The sum is 30

507. What is the output of the following Java program?

```
class BaseTest
{
  void print()
  {
```

```
    System.out.println("BaseTest:print() called");
  }
}
public class Test extends BaseTest
{
  void print()
  {
    System.out.println("Test:print() called");
  }
  public static void main (String args[])
  {
    BaseTest b = new Test();
    b.print();
  }
}
```

Output

Test:print() called

508. What is the output of the following Java program?

```
class Main {
 public static void main(String args[]){
   final int i;
   i = 20;
   System.out.println(i);
 }
}
```

Output

20

509. What is the output of the following Java program?

```
class Person
{
```

```
    public Person()
    {
        System.out.println("Person class constructor called");
    }
}
public class Employee extends Person
{
    public Employee()
    {
        System.out.println("Employee class constructor called");
    }
    public static void main (String args[])
    {
        Employee e = new Employee();
    }
}
```

Output

Person class constructor called

Employee class constructor called

510. What is the output of the following Java program?

```
class Test
{
    int test_a, test_b;
    Test(int a, int b)
    {
    test_a = a;
    test_b = b;
    }
    public static void main (String args[])
    {
```

```
        Test test = new Test();
        System.out.println(test.test_a+" "+test.test_b);
    }
}
```

There is a compiler error in the program because there is a call to the default constructor in the main method which is not present in the class. However, there is only one parameterized constructor in the class Test. Therefore, no default constructor is invoked by the constructor implicitly.

511. What is the output of the following Java program?

```
public class Test
{
    Test(int a, int b)
    {
        System.out.println("a = "+a+" b = "+b);
    }
    Test(int a, float b)
    {
        System.out.println("a = "+a+" b = "+b);
    }
    public static void main (String args[])
    {
        byte a = 10;
        byte b = 15;
        Test test = new Test(a,b);
    }
}
```

The output of the following program is:

a = 10 b = 15

512. What is the output of the following Java program?

```
class Test
{
   int i;
}
public class Main
{
   public static void main (String args[])
   {
      Test test = new Test();
      System.out.println(test.i);
   }
}
```

The output of the program is 0 because the variable i is initialized to 0 internally. As we know that a default constructor is invoked implicitly if there is no constructor in the class, the variable i is initialized to 0 since there is no constructor in the class.

513. What is the output of the following Java program?

```
public class ExceptionHandlingExample {
public static void main(String args[])
{
   try
   {
      int a = 1/0;
      System.out.println("a = "+a);
   }
   catch(Exception e){System.out.println(e);}
   catch(ArithmeticException ex){System.out.println(ex);}
}
}
```

Output

ExceptionHandlingExample.java:10: error: exception ArithmeticException has already been caught

```
        catch(ArithmeticException ex){System.out.println(ex);}
        ^
```

1 error

514. What is the output of the following Java program?

```
class Calculation extends Exception
{
   public Calculation()
   {
      System.out.println("Calculation class is instantiated");
   }
   public void add(int a, int b)
   {
      System.out.println("The sum is "+(a+b));
   }
}
public class Main{
    public static void main(String []args){
       try
       {
          throw new Calculation();
```

```
        }
        catch(Calculation c){
            c.add(10,20);
        }
    }
}
```

Output

Calculation class is instantiated

The sum is 30

515. Write a Java program that prints all the values given at command-line.

Program

```
class A{
public static void main(String args[]){
for(int i=0;i<args.length;i++)
System.out.println(args[i]);
}
}
```

compile by > javac A.java

run by > java A sonoo jaiswal 1 3 abc

Output

sonoo

jaiswal

1

3

abc

516. Write a Java program to sum values of an array?

```
public class Exercise{
public static void main(String[] args) {
int my_array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int sum = 0;
for (int i : my_array)
   sum += i;
System.out.println("The sum is " + sum);
}
}
```

Output: The sum is 55

517. What is the output of the following Java program?

```
public class Main
{
   void a()
   {
      try{
```

```
    System.out.println("a(): Main called");
    b();
    }catch(Exception e)
    {
        System.out.println("Exception is caught");
    }
}
void b() throws Exception
{
 try{
     System.out.println("b(): Main called");
     c();
 }catch(Exception e){
     throw new Exception();
 }
 finally
 {
     System.out.println("finally block is called");
 }
}
void c() throws Exception
```

```
    {
        throw new Exception();
    }

    public static void main (String args[])
    {
        Main m = new Main();
        m.a();
    }
}
```

Output

a(): Main called

b(): Main called

finally block is called

Exception is caught

518. What is the output of the following Java program?

```
public class Calculation
{
    int a;
    public Calculation(int a)
    {
```

```
    this.a = a;
}
public int add()
{
    a = a+10;
    try
    {
        a = a+10;
        try
        {
            a = a*10;
            throw new Exception();
        }catch(Exception e){
            a = a - 10;
        }
    }catch(Exception e)
    {
        a = a - 10;
    }
    return a;
}
```

```
    public static void main (String args[])
    {
        Calculation c = new Calculation(10);
        int result = c.add();
        System.out.println("result = "+result);
    }
}
```

Output

result = 290

519. What is the output of the following Java program?

```
public class Test
public static void main (String args[])
  {
     String a = new String("Sharma is a good player");
     String b = "Sharma is a good player";
     if(a == b)
     {
        System.out.println("a == b");
```

```
    }
    if(a.equals(b))
    {
        System.out.println("a equals b");
    }
 }
```

Output:

a equals b

520. What is the output of the following Java program?

```
class Simple{
 public Simple()
 {
   System.out.println("Constructor of Simple class is invoked");
 }
 void message(){System.out.println("Hello Java");}
}
class Test1{
 public static void main(String args[]){
  try{
```

```
  Class c=Class.forName("Simple");

  Simple s=(Simple)c.newInstance();
  s.message();
  }catch(Exception e){System.out.println(e);
}
 }
}
```

Output

Constructor of Simple class is invoked

Hello Java