**Interview Questions**

**Core Java: -**

**1. What is Java?**

Ans 1.

* Java is a highlevel programming language.
* Object Oriented.
* Multithreaded programming language.
* Aspect oriented and Annotation programming language.
* It is robust.
* Plateform independent.
* Secure

**2. What is JVM?**

Ans 2. It is an abstract machine which provides a run time environment to execute the java bytecode.

**3. What is JDK?**

Ans 3. Java development kit or JDK is a tool that is necessary for compiling the java code. It contains the JRE + other development tool.

**4. What is JRE?**

Ans 4. The Java Runtime Environment is the implementation of JVM in which the byte code is executed.

**5. Types of Memory allocated by JVM?**

Ans 5.

* Class Area
* Heap
* Stack
* Program Counter
* Native Method Stack

**6. What is JIT?**

Ans 6.

* It improves the performance.
* Compiles the part of bytecode with similar functionality at the same time.
* Reduces the amount of time needed for compilation.

**7. What if I write static public void instead of public static void?**

Ans 7. Order of specifiers doesnot matter in java.

**8. What is the default value of the local variables?**

Ans 8. The local variables are not initialized to any value.

**9. What are the various access specifiers in Java?**

Ans 9. 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.

**OOPs Concept**

**10. What is object-oriented paradigm?**

Ans 10. It is a programming paradigm based on objects having data and methods defined in the class to which it belongs. Object-oriented paradigm aims to incorporate the advantages of modularity and reusability. Objects are the instances of classes which interacts with one another to design applications and programs. There are the following features of the object-oriented paradigm.

* Follows the bottom-up approach in program design.
* Focus on data with methods to operate upon the object's data
* Includes the concept like Encapsulation and abstraction which hides the complexities from the user and show only functionality.
* Implements the real-time approach like inheritance, abstraction, etc.
* The examples of the object-oriented paradigm are C++, Simula, Smalltalk, Python, C#, etc

**11. What is an object?**

Ans 11. 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.

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

Ans 12. All object references are initialized to null in Java.

**13.  What is the constructor?**

Ans 13. 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.

**14. Does constructor return any value?**

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

**15. Is constructor inherited?**

Ans 15. No, The constructor is not inherited.

**16. Can you make a constructor final?**

Ans 16. No, the constructor can't be final.

**17. What is the static variable?**

Ans 17. The static variable is used to refer to the common property of all objects (that is not unique for each object), e.g., The company name of employees, college name of students, etc. Static variable gets memory only once in the class area at the time of class loading. Using a static variable makes your program more memory efficient (it saves memory). Static variable belongs to the class rather than the object.

**18.  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.

**19. 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.

**20. 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.

**21. Can we override the static methods?**

No, we can't override static methods.

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

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

**23. 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.

**24.  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.

**25. 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.

**26. 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.

**27. 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.

**28. 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.

**28. 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.

**29. 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.

**30.  Which class is the superclass for all the classes?**

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

**31. 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.

**32. 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.

**33. 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.

**34. 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.

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

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

**36. 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.

* By Changing the number of arguments
* By Changing the data type of arguments

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

**37. 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.

**38. 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). Consider the following example.

**39.  Can we overload the main() method?**

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

**40. What is method overloading with type promotion?**

By Type promotion is method overloading, we mean that one data type can be promoted to another implicitly if no exact matching is found.

**41. 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.

**41. 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.

**42. 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.

**43. Can we override the overloaded method?**

Yes.

**44. 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.

**45.  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.

**46. What is covariant return type?**

Now, since java5, it is possible to override any method by changing the return type if the return type of the subclass overriding method is subclass type. It is known as covariant return type. The covariant return type specifies that the return type may vary in the same direction as the subclass.

**47. 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. In other words, we can say that the final variable once assigned to a value, can never be changed after that. The final variable which is not assigned to any value can only be assigned through the class constructor.

**48. What is the final method?**

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

**49. Can you declare the main method as final?**

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

**50. 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.

**51.  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.

**52. 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.

**53.  What is Runtime Polymorphism?**

Runtime polymorphism or dynamic method dispatch is a process in which a call to an overridden method is resolved at runtime rather than at compile-time. In this process, an overridden method is called through the reference variable of a superclass.

**54. 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.

**55. 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

**56. What is the difference between abstraction and encapsulation**?

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

**57. 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.

**58. 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.

**59. 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

**60.  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.

**61. 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.

**62.  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.

**63. 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.

**64. 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.

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

No, they are implicitly public.

**66. 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.

**67. 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.

**68. 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.

**69. 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.

**70. 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.

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

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

**72. 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.

**73. 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.

**74. 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.

**75. 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.

**76. What is the base class for Error and Exception?**

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

**77. 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.

**78.  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).

**79. 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.

**80. 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).

**81.  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.

**82. What is the meaning of immutable regarding String?**

The simple meaning of immutable is unmodifiable or unchangeable. In Java, String is immutable, i.e., once string object has been created, its value can't be changed. Consider the following example for better understanding.

**83. 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.

**84. How many ways can we create the string object?**

**1) String Literal**

Java String literal is created by using double quotes. For Example:

**String s="welcome";**

Each time you 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. String objects are stored in a special memory area known as the **string constant pool** For example:

**String s1="Welcome";**

**String s2="Welcome";//It doesn't create a new instance**

**2) By new keyword**

**String s=new String("Welcome");//creates two objects and one reference variable**

In such case, JVM will create a new string object in normal (non-pool) heap memory, and the literal "Welcome" will be placed in the constant string pool. The variable s will refer to the object in a heap (non-pool).

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

1. String s1="Welcome";
2. String s2="Welcome";
3. String s3="Welcome";

Only one object will be created using the above code because strings in Java are immutable.

**86. 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).

**87. 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.

**88. 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.

**89. 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

**90. 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.

**91. 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.

**92. 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.

**93. 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.

**94. 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.

**95. 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. Consider the following example for the better understanding of how the gc() method invoke the garbage collector.

**96. 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.

**97. What kind of thread is the Garbage collector thread?**

Daemon thread.

**98. 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

**99. 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. Consider the following example of writing a byte into a file.

100. 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.

**101. 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.

**102.  How can you make a class serializable in Java?**

A class can become serializable by implementing the Serializable interface.

**103. 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.

**104. 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.

**105.  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

**106. What are wrapper classes in Java?**

Wrapper classes convert the Java primitives into the reference types (objects). Every primitive data type has a class dedicated to it. These are known as wrapper classes because they “wrap” the primitive data type into an object of that class. Refer to the below image which displays different primitive type, wrapper class and constructor argument.

**107. What is the difference between equals() and == in Java?**

Equals() method is defined in Object class in Java and used for checking equality of two objects defined by business logic.

“==” or equality operator in Java is a binary operator provided by Java programming language and used to compare primitives and objects. *public boolean equals(Object o)* is the method provided by the Object class. The default implementation uses == operator to compare two objects. For example: method can be overridden like String class. equals() method is used to compare the values of two objects.

**107. What is multi-threading?**

Ans: Multi threading is a programming concept to run multiple tasks in a concurrent manner within a single program. Threads share same process stack and running in parallel. It helps in performance improvement of any program.

**108. What are the two ways of implementing multi-threading in Java?**

Ans: Multi threaded applications can be developed in Java by using any of the following two methodologies:

1. By using Java.Lang.Runnable Interface. Classes implement this interface to enable multi threading. There is a Run() method in this interface which is implemented.

2. By writing a class that extend Java.Lang.Thread class.

**109. When a lot of changes are required in data, which one should be a preference to be used? String or StringBuffer?**

Ans: Since StringBuffers are dynamic in nature and we can change the values of StringBuffer objects unlike String which is immutable, it’s always a good choice to use StringBuffer when data is being changed too much. If we use String in such a case, for every data change a new String object will be created which will be an extra overhead.

**110.  In multi-threading how can we ensure that a resource isn’t used by multiple threads simultaneously?**

Ans: In multi-threading, access to the resources which are shared among multiple threads can be controlled by using the concept of synchronization. Using synchronized keyword, we can ensure that only one thread can use shared resource at a time and others can get control of the resource only once it has become free from the other one using it.

**111. Difference between throw and throws keyword.**Ans. When a program is unable to produce the required output, ‘throw’ keyword is used. It helps us to create an exception and interrupt the flow of the program.  
‘Throws’ keyword is used to signal a probable exception in a program may occur when a method called is executed.

**112. Explain 5 keywords used in exception handling.**Ans. **Try**– Exception is handled by writing the code inside the try block that might throw an exception.

**Catch**– The exception handling code is written in the catch block.  
  
**Throw**– It is used by the user to create an exception if the code does not run in the desired way.  
  
**Throws**– When we are aware of the checked exceptions and let the caller program know about those, the throws keyword is used before that exception.  
  
**Finally**– This block always gets executed even if an exception is thrown. It is used with the try-catch blocks.

**113. What is synchronization in java?**Ans. When multiple threads(instructions) try to access the same resource, errors are bound to happen. Using synchronized blocks in java, you can control the access to these multiple threads. This is called synchronization. These synchronized blocks can be identified by the synchronized keyword.

**114. What is the difference between array and arrayList?**Ans. **Array-**  
1. It is a dynamic object and holds similar values.  
2. It is static in size, meaning the size cannot be manipulated once created.  
3. Can store both objects and primitives.  
4. Multidimensional.

**ArrayList-**1. It is a class of Java collections framework and comes under Java,util package.  
2. It is dynamic in size. Therefore, can be resized according to the need.  
3. Cannot store primitives.  
4. It is always of single-dimension.

**115. What is a list in Java?**Ans. The list is an interface in Java in which objects can be stored in an ordered way(indexed) and duplicate and null values can also be stored. ArrayList, Linked List, vector, and stack are implementation classes of the List.

**116. What is the Collection interface in Java?**Ans. It is a framework which acts as a base to store and manipulate groups of objects. Classes like ArrayList, LinkedList, Vector and interfaces like queue, list, set come under it.

**117.   What is a hash map?**Ans. Hashmap is an implementation of map interface. The data is stored in pairs in the form of key, value. The key acts as an index to another object(value).

**118. 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.

**119. 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.

**120. 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.

**121. 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.

**JDBC Interview Questions**

**122. What is JDBC?**

JDBC is a Java API that is used to connect and execute the query to the database. JDBC API uses JDBC drivers to connect to the database. JDBC API can be used to access tabular data stored into any relational database.

**123. What is JDBC Driver?**

JDBC Driver is a software component that enables Java application to interact with the database. There are 4 types of JDBC drivers:

1. **JDBC-ODBC bridge driver:** The JDBC-ODBC bridge driver uses the ODBC driver to connect to the database. The JDBC-ODBC bridge driver converts JDBC method calls into the ODBC function calls. This is now discouraged because of the thin driver. It is easy to use and can be easily connected to any database.
2. **Native-API driver (partially java driver):** The Native API driver uses the client-side libraries of the database. The driver converts JDBC method calls into native calls of the database API. It is not written entirely in Java. Its performance is better than JDBC-ODBC bridge driver. However, the native driver must be installed on each client machine.
3. **Network Protocol driver (fully java driver):** The Network Protocol driver uses middleware (application server) that converts JDBC calls directly or indirectly into the vendor-specific database protocol. It is entirely written in Java. There is no requirement of the client-side library because of the application server that can perform many tasks like auditing, load balancing, logging, etc.
4. **Thin driver (fully java driver):** The thin driver converts JDBC calls directly into the vendor-specific database protocol. That is why it is known as the thin driver. It is entirely written in Java language. Its performance is better than all other drivers however these drivers depend upon the database.

**124. What are the JDBC API components?**

The java.sql package contains following interfaces and classes for JDBC API.

**Interfaces:**

* **Connection:** The Connection object is created by using getConnection() method of DriverManager class. DriverManager is the factory for connection.
* **Statement:** The Statement object is created by using createStatement() method of Connection class. The Connection interface is the factory for Statement.
* **PreparedStatement:** The PrepareStatement object is created by using prepareStatement() method of Connection class. It is used to execute the parameterized query.
* **ResultSet:** The object of ResultSet maintains a cursor pointing to a row of a table. Initially, cursor points before the first row. The executeQuery() method of Statement interface returns the ResultSet object.
* **CallableStatement:** CallableStatement interface is used to call the stored procedures and functions. We can have business logic on the database through the use of stored procedures and functions that will make the performance better because these are precompiled. The prepareCall() method of Connection interface returns the instance of CallableStatement.

**Classes:**

* **DriverManager:** The DriverManager class acts as an interface between the user and drivers. It keeps track of the drivers that are available and handles establishing a connection between a database and the appropriate driver. It contains several methods to keep the interaction between the user and drivers.
* **Blob:** Blob stands for the binary large object. It represents a collection of binary data stored as a single entity in the database management system.
* **Clob:** Clob stands for Character large object. It is a data type that is used by various database management systems to store character files. It is similar to Blob except for the difference that BLOB represent binary data such as images, audio and video files, etc. whereas Clob represents character stream data such as character files, etc.
* **SQLException** It is an Exception class which provides information on database access errors.

**125. What is the return type of Class.forName() method?**

The Class.forName() method returns the object of java.lang.Class object.

**126.  What are the benefits of PreparedStatement over Statement?**

The benefits of using PreparedStatement over Statement interface is given below.

* The PreparedStatement performs faster as compare to Statement because the Statement needs to be compiled everytime we run the code whereas the PreparedStatement compiled once and then execute only on runtime.
* PreparedStatement can execute Parameterized query whereas Statement can only run static queries.
* The query used in PreparedStatement is appeared to be similar every time. Therefore, the database can reuse the previous access plan whereas, Statement inline the parameters into the String, therefore, the query doesn't appear to be same everytime which prevents cache reusage.

**127. What is the role of the JDBC DriverManager class?**

The DriverManager class acts as an interface between user and drivers. It keeps track of the drivers that are available and handles establishing a connection between a database and the appropriate driver. The DriverManager class maintains a list of Driver classes that have registered themselves by calling the method DriverManager.registerDriver().

**128. What are the functions of the JDBC Connection interface?**

The **Connection interface** maintains a session with the database. It can be used for transaction management. It provides factory methods that return the instance of Statement, PreparedStatement, CallableStatement, and DatabaseMetaData.

**129. What does the JDBC ResultSet interface?**

The ResultSet object represents a row of a table. It can be used to change the cursor pointer and get the information from the database. By default, ResultSet object can move in the forward direction only and is not updatable. However, we can make this object to move the forward and backward direction by passing either TYPE\_SCROLL\_INSENSITIVE or TYPE\_SCROLL\_SENSITIVE in createStatement(int, int) method.

**130. Which interface is responsible for transaction management in JDBC?**

The **Connection interface** provides methods for transaction management such as commit(), rollback() etc.

**131.  What are CLOB and BLOB data types in JDBC?**

**BLOB:** Blob can be defined as the variable-length, binary large object which is used to hold the group of Binary data such as voice, images, and mixed media. It can hold up to 2GB data on MySQL database and 128 GB on Oracle database. BLOB is supported by many databases such as MySQL, Oracle, and DB2 to store the binary data (images, video, audio, and mixed media).

**CLOB:** Clob can be defined as the variable-length, character-large object which is used to hold the character-based data such as files in many databases. It can hold up to 2 GB on MySQL database, and 128 GB on Oracle Database. A CLOB is considered as a character string.

**SQL**

**132.  What is SQL?**

SQL stands for the Structured Query Language. It is the standard language used to maintain the relational database and perform many different data manipulation operations on the data. SQL was initially invented in 1970. It is a database language used for database creation, deletion, fetching and modifying rows, etc. sometimes, it is pronounced as 'sequel.' We can also use it to handle organized data comprised of entities (variables) and relations between different entities of the data.

**133. What are the subsets of SQL?**

The following are the four significant subsets of the SQL:

* **Data definition language (DDL):** It defines the data structure that consists of commands like CREATE, ALTER, DROP, etc.
* **Data manipulation language (DML):** It is used to manipulate existing data in the database. The commands in this category are SELECT, UPDATE, INSERT, etc.
* **Data control language (DCL):** It controls access to the data stored in the database. The commands in this category include GRANT and REVOKE.
* **Transaction Control Language (TCL):** It is used to deal with the transaction operations in the database. The commands in this category are COMMIT, ROLLBACK, SET TRANSACTION, SAVEPOINT, etc.

**134. What is the purpose of DDL Language?**

DDL stands for Data definition language. It is the subset of a database that defines the data structure of the database when the database is created. **For example,** we can use the DDL commands to add, remove, or modify tables. It consists of the following commands: CREATE, ALTER and DELETE database objects such as schema, tables, indexes, view, sequence, etc.

**135. What is the purpose of DML Language?**

Data manipulation language makes the user able to retrieve and manipulate data in a relational database. The DML commands can only perform read-only operations on data. We can perform the following operations using DDL language:

* Insert data into the database through the INSERT command.
* Retrieve data from the database through the SELECT command.
* Update data in the database through the UPDATE command.
* Delete data from the database through the DELETE command.

**136. What is the purpose of DCL Language?**

Data control language allows users to control access and permission management to the database. It is the subset of a database, which decides that what part of the database should be accessed by which user at what point of time. It includes two commands, GRANT and REVOKE.

**GRANT:** It enables system administrators to assign privileges and roles to the specific user accounts to perform specific tasks on the database.

**REVOKE:** It enables system administrators to revoke privileges and roles from the user accounts so that they cannot use the previously assigned permission on the database.

**137.  What is a primary key?**

A primary key is a field or the combination of fields that uniquely identify each record in the table. It is one of a special kind of unique key. If the column contains a primary key, it cannot be null or empty. A table can have duplicate columns, but it cannot have more than one primary key. It always stores unique values into a column. **For example,** the ROLL Number can be treated as the primary key for a student in the university or college.

**138. What is a foreign key?**

The foreign key is used to link one or more tables together. It is also known as the referencing key. A foreign key is specified as a key that is related to the primary key of another table. It means a foreign key field in one table refers to the primary key field of the other table. It identifies each row of another table uniquely that maintains the referential integrity. The primary key-foreign key relationship is a very crucial relationship as it maintains the ACID properties of the database sometimes. It also prevents actions that would destroy links between the child and parent tables.

**139. What is a unique key?**

A unique key is a single or combination of fields that ensure all values stores in the column will be unique. It means a column cannot stores duplicate values. This key provides uniqueness for the column or set of columns. **For example,** the email addresses and roll numbers of student's tables should be unique. It can accept a null value but only one null value per column. It ensures the integrity of the column or group of columns to store different values into a table.

**140. What is the difference between a primary key and a unique key?**

The primary key and unique key both are essential constraints of the SQL. The main difference among them is that the primary key identifies each record in the table. In contrast, the unique key prevents duplicate entries in a column except for a NULL value.

**141. What is the SQL query to display the current date?**

There is a built-in function in SQL called GetDate(), which is used to return the current timestamp.

**142. Which are joins in SQL? Name the most commonly used SQL joins?**

SQL joins are used to retrieve data from multiple tables into a meaningful result set. It is performed whenever you need to fetch records from two or more tables. They are used with SELECT statement and join conditions.

The following are the most commonly used joins in SQL:

* INNER JOIN
* LEFT OUTER JOIN
* RIGHT OUTER JOIN

**143. What are the different types of joins in SQL?**

Joins are used to merge two tables or retrieve data from tables. It depends on the relationship between tables. According to the ANSI standard, the following are the different types of joins used in SQL:

* INNER JOIN
* SELF JOIN
* LEFT OUTER JOIN
* RIGHT OUTER JOIN
* FULL OUTER JOIN
* CROSS JOIN

**144. What is INNER JOIN in SQL?**

Inner join returns only those records from the tables that match the specified condition and hides other rows and columns. In simple words, it fetches rows when there is at least one match of rows between the tables is found. INNER JOIN keyword joins the matching records from two tables. It is assumed as a default join, so it is optional to use the INNER keyword with the query.

**145. What is the Right JOIN in SQL?**

The Right join is used to retrieve all rows from the right-hand table and only those rows from the other table that fulfilled the join condition. It returns all the rows from the right-hand side table even though there are no matches in the left-hand side table. If it finds unmatched records from the left side table, it returns a Null value. This join is also known as Right Outer Join.

**146. What is Left Join in SQL?**

The Left Join is used to fetch all rows from the left-hand table and common records between the specified tables. It returns all the rows from the left-hand side table even though there are no matches on the right-hand side table. If it will not find any matching record from the right side table, then it returns null. This join can also be called a Left Outer Join.

**147. What is Full Join in SQL?**

The Full Join results from a combination of both left and right join that contains all the records from both tables. It fetches rows when there are matching rows in any one of the tables. This means it returns all the rows from the left-hand side table and all the rows from the right-hand side tables. If a match is not found, it puts NULL value. It is also known as FULL OUTER JOIN.

**148. What is a constraint?**

The constraint is used to specify the rule and regulations that allows or restricts what values/data will be stored in the table. It ensures data accuracy and integrity inside the table. It enforces us to store valid data and prevents us from storing irrelevant data. If any interruption occurs between the constraint and data action, the action is failed. Some of the most commonly used constraints are NOT NULL, PRIMARY KEY, FOREIGN KEY, AUTO\_INCREMENT, UNIQUE KEY, etc.

**HTML &CSS**

**149.  What is HTML?**

HTML stands for Hyper Text Markup Language. It is a language of World Wide Web. It is a standard text formatting language which is used to create and display pages on the Web. It makes the text more interactive and dynamic. It can turn text into images, tables, links.

**150. What are Tags?**

HTML tags are composed of three things: an opening tag, content and ending tag. Some tags are unclosed tags.

HTML documents contain two things:

* content, and
* tags

When a web browser reads an HTML document, the browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties.

**151. Do all HTML tags have an end tag?**

No. There are some HTML tags that don't need a closing tag. For example: <image> tag, <br> tag.

**152. What is formatting in HTML?**

The HTML formatting is a process of format the text for a better look and feel. It uses different tags to make text bold, italicized, underlined.

**153. Which HTML tag is used to display the data in the tabular form?**

The **HTML table tag** is used to display data in tabular form (row \* column). It also manages the layout of the page, e.g., header section, navigation bar, body content, footer section.

**154. How to insert a copyright symbol on a browser page?**

You can insert a copyright symbol by using &copy; or &#169; in an HTML file.

**155.  How to create a nested webpage in HTML?**

The HTML iframe tag is used to display a nested webpage. In other words, it represents a webpage within a webpage. The HTML <iframe> tag defines an inline frame.

**156. Does a <!DOCTYPE html> tag is a HTML tag?**

No, the <!DOCTYPE html> declaration is not an HTML tag. There are many type of HTML e.g. HTML 4.01 Strict, HTML 4.01 Transitional, HTML 4.01 Frameset, XHTML 1.0 Strict, XHTML 1.0 Transitional, XHTML 1.0 Frameset, XHTML 1.1 etc. So, <!DOCTYPE html> is used to instruct the web browser about the HTML page.

**157. What is CSS?**

CSS stands for Cascading Style Sheet. It is a popular styling language which is used with HTML to design websites. It can also be used with any XML documents including plain XML, SVG, and XUL

**158. What is the origin of CSS?**

SGML (Standard Generalized Markup Language) is the origin of CSS. It is a language that defines markup languages.

**159.  How can you integrate CSS on a web page?**

There are three methods to integrate CSS on web pages.

1. Inline method - It is used to insert style sheets in HTML document
2. Embedded/Internal method - It is used to add a unique style to a single document
3. Linked/Imported/External method - It is used when you want to make changes on multiple pages.

**160. What is a CSS selector?**

It is a string that identifies the elements to which a particular declaration apply. It is also referred as a link between the HTML document and the style sheet. It is equivalent of HTML elements. There are several different types of selectors in CSS: -

* CSS Element Selector
* CSS Id Selector
* CSS Class Selector
* CSS Universal Selector
* CSS Group Selector

**JAVASCRIPT & JQUERY**

**161. What is JavaScript?**

**JavaScript** is *a scripting language*. It is different from Java language. It is object-based, lightweight, cross-platform translated language. It is widely used for client-side validation.

**162. List some features of JavaScript.**

Some of the features of JavaScript are:

* Lightweight
* Interpreted programming language
* Good for the applications which are network-centric
* Complementary to Java
* Complementary to HTML
* Open source

Cross-platform

**163.  List some of the advantages of JavaScript.**

Some of the advantages of JavaScript are:

* Server interaction is less
* Feedback to the visitors is immediate
* Interactivity is high
* Interfaces are richer

**164. List some of the disadvantages of JavaScript.**

Some of the disadvantages of JavaScript are:

* No support for multithreading
* No support for multiprocessing
* Reading and writing of files is not allowed
* No support for networking applications.

**165. Define a named function in JavaScript.**

The function which has named at the time of definition is called a named function. For example

**function msg()**

**{    document.writeln("Named Function");**

**}**

**msg();**

**166.  Is JavaScript case sensitive language?**

Yes, JavaScript is a case sensitive language.

**167. How to write a comment in JavaScript?**

There are two types of comments in JavaScript.

1. Single Line Comment: It is represented by // (double forward slash)
2. Multi-Line Comment: Slash represents it with asterisk symbol as /\* write comment here \*/

**168. How to create a function in JavaScript?**

To create a function in JavaScript, follow the following syntax.

**function function\_name(){**

**//function body**

**}**

**169. How to create objects in JavaScript?**

There are 3 ways to create an object in JavaScript.

1. By object literal
2. By creating an instance of Object
3. By Object Constructor

**emp={id:102,name:"Rahul Kumar",salary:50000}**

**170. What is jQuery?**

jQuery is a fast, lightweight, feature-rich client-side JavaScript library. It is cross-platform and supports different types of browsers. It has provided a much-needed boost to JavaScript. Before jQuery, JavaScript codes were lengthy and bigger, even for smaller functions. It makes a website more interactive and attractive.

**171.  Is jQuery a programming language?**

jQuery is not a programming language but a well-written JavaScript code. It is used to traverse documents, event handling, Ajax interaction, and Animation.

**172. What is the difference between JavaScript and jQuery?**

The simple difference is that JavaScript is a language while jQuery is a built-in library built for JavaScript. jQuery simplifies the use of JavaScript language.

**173.  Why do we use jQuery?**

* It is very easy to learn and use.
* It is used to develop browser compatible web applications.
* It improves the performance of an application.
* It is very fast and extensible.
* It facilitates you to write minimal lines of codes for UI related functions.
* It provides cross-browser support.

**174. What is $() in jQuery library?**

The $() function is an alias of jQuery() function. It is used to wrap any object into jQuery object which later facilitates you to call the various method defined jQuery object. You can pass a selector string to $() function, and it returns a jQuery object which contains an array of all matched DOM elements.

Syntax:

**$(document).ready(function() {**

**$("p").css("background-color", "pink");**

**});**

175. What is the use of val() method in JQuery?

The jQuery val() method is used:

* To get the current value of the first element in the set of matched elements.
* To set the value of every matched element.

**Servlet**

**176. How many objects of a servlet is created?**

Only one object at the time of first request by servlet or web container.

**177. What is the life-cycle of a servlet?**

1. Servlet is loaded
2. servlet is instantiated
3. servlet is initialized
4. service the request
5. servlet is destroyed

**178. Who is responsible to create the object of servlet?**

The web container or servlet container.

**179. When servlet object is created?**

At the time of first request.

**180. What is servlet collaboration?**

When one servlet communicates to another servlet, it is known as servlet collaboration. There are many ways of servlet collaboration:

* RequestDispacher interface
* sendRedirect() method etc.

**181. What is the purpose of RequestDispatcher Interface?**

The RequestDispacher interface provides the facility of dispatching the request to another resource it may be html, servlet or jsp. This interceptor can also be used to include the content of antoher resource.

**182. Can you call a jsp from the servlet?**

Yes, one of the way is RequestDispatcher interface for example:

**RequestDispatcher rd=request.getRequestDispatcher("/login.jsp");**

**rd.forward(request,response);**

**183. What is difference between ServletConfig and ServletContext?**

The container creates object of ServletConfig for each servlet whereas object of ServletContext is created for each web application.

**184. What is Session Tracking?**

**Session** simply means a particular interval of time.

Session Tracking is a way to maintain state of an user.Http protocol is a stateless protocol.Each time user requests to the server, server treats the request as the new request.So we need to maintain the state of an user to recognize to particular user.

**185. What are Cookies?**

A cookie is a small piece of information that is persisted between the multiple client requests. A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number.

**186. What is difference between Cookies and HttpSession?**

Cookie works at client side whereas HttpSession works at server side.

**187. What is filter?**

A filter is an object that is invoked either at the preprocessing or postprocessing of a request. It is pluggable.

**188. What is the disadvantage of cookies?**

It will not work if cookie is disabled from the browser.

**189. What is the use of welcome-file-list?**

It is used to specify the welcome file for the project.

**190. What is the use of attribute in servlets?**

Attribute is a map object that can be used to set, get or remove in request, session or application scope. It is mainly used to share information between one servlet to another.

**JSP**

**191.  What is JSP?**

Java Server Pages technology (JSP) is a server-side programming language used to create a dynamic web page in the form of HyperText Markup Language (HTML). It is an extension to the servlet technology.

A JSP page is internally converted into the servlet. JSP has access to the entire family of the Java API including JDBC API to access enterprise database. Hence, Java language syntax has been used in the java server pages (JSP). The JSP pages are more accessible to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

**192. List out some advantages of using JSP.**

* Better performance.
* The compilation of JSP is done before it is processed by the server which eradicates the need for loading of interpreter and code script each time.
* JSP has access to all-powerful enterprises.  
  Easy to maintain: JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic.
* JSP can also be used in combination with servlets.

**193. Give the syntax for JSP comments.**

The syntax for JSP comments is:

1. **<%-- Comment --%>**

**194. How can we handle the exceptions in JSP?**

There are two ways to perform exception handling, one is by the errorPage element of page directive, and second is by the error-page element of the web.xml file.

**195.  How can we forward the request from JSP page to the servlet?**

Yes of course! With the help of "forward action" tag, but we need to give the URL-pattern of the servlet.

**196.  How is JSP used in the MVC model?**

JSP is usually used for presentation in the MVC pattern (Model View Controller ), i.e., it plays the role of the view. The controller deals with calling the model and the business classes which in turn get the data, and this data is then presented to the JSP for rendering on to the client.

**197.  What are context initialization parameters?**

Context initialization parameters are specified by the <context-param> in the web.xml file, and these are initialization parameter for the whole application and not specific to any servlet or JSP.

**198. What is the purpose of <jsp:useBean>?**

The jsp:useBean action searches for the existence of the object with specified name. If not found, it creates one.

**199. What is JSTL?**

JSP Standard Tag Library is a library of predefined tags that ease the development of JSP.

**200.  List the various action tags used in JSP.**

Following are the list of various action tags used in JSP:

* jsp:forward: This action tag forwards the request and response to another resource.
* jsp:include: This action tag is used to include another resource.
* jsp:useBean: This action tag is used to create and locates bean object.
* jsp:setProperty: This action tag is used to set the value of the property of the bean.
* jsp:getProperty: This action tag is used to print the value of the property of the bean.
* jsp:plugin: This action tag is used to embed another component such as the applet.
* jsp:param: This action tag is used to set the parameter value. It is used in forward and includes mostly.
* jsp:fallback: This action tag can be used to print the message if the plugin is working.

**201. What is Expression language in JSP?**

The Expression Language (EL) simplifies the accessibility of data stored in the Java Bean component, and other objects like request, session, application.

There are many implicit objects, operators and reserve words in EL.