CORE JAVA INTERVIEW QUESTIONS

1) Who Developed Java Programming Language?

James Gosling

2) Why is java so popular?

Java is Platform Independent, Distributed, High Performance Programming Language

3) What is platform independent?

If an application runs everywhere without dependent on OS, then it is platform independent.

4) What is bytecode?

It is one type of highly optimized code which is understandable to the JVM

5) Compare JDK vs JRE vs JVM

To develop, compile and run Java Programs we require JDK

To only run Java Programs, JRE is enough

JVM's main responsibility is converting Byte code instructions to machine understandable format.

6) What is a Token? how many types?

A smallest individual unit in a program is known as Token. Tokens are of 5 types.

- Keywords
- Identifiers
- Literals
- Operators
- Separators

7) Define Datatype and its types?

Datatype is the one that is used to specify two things about a variable

the quantity of memory required for the variable

the type of value stored into a variable.

Datatypes are classified into 4 categories:

- Integral datatypes
  - byte short int long
- Real or Floating datatypes
  - float double
- Character Datatype
  - char

- Boolean datatype
    - boolean

8) What are different types of Operators in Java?

- Unary, Binary, Ternary Operators
- Arithmetic Operators
- Relational Operators
- Logical Operators
- Conditional Operators
- Increment & Decrement Operators
- Assignment Operators
- Bitwise Operators
- Boolean Operators
- Special Operators
    - Dot Operator
    - New Operator
    - Instanceof Operator
    - Cast Operator

9) What is casting?

Converting from one datatype to another datatype is called Casting

10) What is implicit casting?

Converting from smaller datatype to larger datatype is called implicit typecasting. It will be done by java directly.

11) What is explicit casting?

Converting from larger datatype to smaller datatype is called explicit typecasting. It should be done explicitly by using cast operator

12) What is an Array? Explain Different types of Arrays?

An Array is an ordered collection of fixed number of homogeneous or similar datatype elements.

- One Dimensional
- MultiDimensional (2D 3D)
- Jagged Arrays/Ragged Arrays / Irregular MultiDimensional Arrays
- Anonymous Arrays

13) Explain Arrays class methods?

- binarySearch()
- sort()
- fill()
- copyOf()
- copyOfRange()
- hashCode()
- toString()
- equals()
- deepEquals()

14) Difference between length and length()?

- length is the variable which is used to find the length of an array
- length() is used to find the length of the string

15) What is Jagged Array?

- IrregularMultiDimensional Arrays.
- where the rows and columns are not in equal size

16) What is a class?

Class is a blueprint for an object. class contains variables and methods

A class is a model for creating objects

(or)

A class is a template from which we can create objects.

(or)

A class is a collection of objects of similar type

(or)

A class is a blueprint for an object. It tells the virtual machine how to make an object of that particular type

(or)

A class is used to construct objects

(or)

A class contains variables and methods

- The variables in a class are known as instance variables
- The methods in a class are known as instance methods

17) What is an object?

Object is an instance of a class

An Object knows things and does things

Things an object knows about itself are called instance variables. They represent the state of an object

Things an object (does) can do things are called instance methods. They represent the behavior of an object

18) What is state of an object?

instance variables

19) What is behaviour of an object?

- instance methods

20) What are different types of Variables in Java?

- Local variables
- Instance Variables
- Static Variables
- Final Variables
- Transient Variables
- Volatile Variables
- public static final variables

21) What are different type of Methods in java?

- 23 different types of methods

22) Explain inheritance with examples?

- Acquiring the properties of objects of one class to another class is known as Inheritance.
- existing class is known as Super class or parent class
- newly created class is known as Sub class or child class

23) What is method overloading?

- Defining more than one method with same name same return type or different type but compulsory difference in arguments is known as Method Overloading.
- Overloading can be achieved in a single class or same class.

24) What is method overriding?

- Defining more than one method with same name same return type and same arguments but one in super class and another one in sub class is known as Method Overriding.
- this can be achieved in super and sub classes.

25) What is method hiding?

- Defining more than one static method with same name same return type and same arguments but one in super class and another one in sub class is known as Method Overriding.

26) Can u achieve method overriding on static methods?

- No

27) What is Upcasting ?

- When super class reference variable refers to the sub class object then it is known as upcasting. In this upcasting you can access only super class methods, and if any super class methods overridden in the sub class then sub class overridden methods can also be accessed. but we cannot access sub class own methods.

28) Can super class reference variable hold an object of sub class?

- yes

29) Is multiple inheritance allowed in java?

- yes. by using interfaces in 2 ways:

- A Class implements multiple interfaces
- An interface can extends multiple interfaces

30) What is a constructor?

- A constructor is a special method which can be used to initialize the instance variables of the class or to initialize the object of the class.

31) What is a default constructor?

- If a constructor don't have any arguments then it is known as default constructor

32) How do you call a super class constructor from sub class constructor?

- using super()
- or
- super(arg list)

33) What is the use of this()?

- this() can be used to call the current class no argument constructor.

34) Can a constructor be called directly from a method?

- yes

35) Is a super class constructor called even when there is no explicit call from a sub class constructor?

- yes

36) What is polymorphism?

- An ability to take more than one form is known as Polymorphism.

37) What is the use of instanceof operator in java?

- It is used to check whether the given object belongs to the given class or not. it returns true if it belongs to the class otherwise returns false.

38) What is encapsulation?

Wrapping up of data and methods together is known as encapsulation. best example is class.

Encapsulation is hiding the implementation of a class behind a well defined interface.

Encapsulation helps us to change implementation of a class without breaking other code.

39) What is interface?

An interface is just like a class, which contains constant variables and public abstract methods.

40) How do u define an interface?

```
interface interfacename
        {
            constant variables
            abstract methods
        }
```

41) How do you implement interface?

- class implements interface

42) What else u know about interface?

- It is a mechanism to achieve fully abstraction.
- Java interface also represents IS-A relationship
- Interface is also called 100% pure Abstract class.
- Multiple Inheritance is achieved in java using interfaces
- Interface is just like a class but it contains only abstract methods and constant variables upto JDK 7.
- We should not create object for interface because it contains incomplete methods or abstract methods.
- but we can create implementation classes for interface.
- If a class provides method body or implementation for abstract methods of interface then that class is known as implementation class of that interface.
- By default, all the methods in interface are public and abstract.
- By default, the variables in interface are public static final.

43) Can u extend an interface?

- yes

44) Can a class extend multiple interfaces?

A class can implements multiple interfaces

45) What is an abstract class?

An abstract class is a class which contains both abstract methods and non-abstract methods.

46) When do u use an abstract class?

Separate ideas with implementation

47) How do you define an abstract method?

abstract returntype methodname();

48) Compare abstract class and interface?

We cant achieve 100% abstraction using abstract classes but we can achieve 100% using interfaces.

Multiple inheritance cannot be achieved using abstract classes but we can achieve inheritance using interfaces.

49) What is an inner class?

A class inside another class is known as inner class

50) What is a static inner class?

A class declared directly inside another class and declared as static.

51) Can u create a inner class inside a method?

Yes. local inner class

52) What is an anonymous class?

Anonymous class does not have any name.

53). What is private access modifier?

Private variables and methods we can use it only with in the class where it is declared.

54). What is default or package access modifier?

If no modifier is used it is called default access modifier

55) What is protected access modifier?

When we declare any variable or method as protected that we can use it any where in any class within the package, but outside of the package only in sub classes we can use it.

56) What is public access modifier?

we can use everywhere, in any package.

57) What access types of variables can be accessed from a class in same package?

default, public, protected

58). What access types of variables can be accessed from a class in different package?

public

59) What access types of variables can be accessed from a sub class in same package?

default, protected, public

60) What access types of variables can be accessed from a sub class in different package?

protected, public

61) What is the use of final modifier on a variable?

Final variable value is fixed cannot be changed

62) What is the use of final modifier on a class?

Final classes cannot be inherited

63) What is the use of final modifier on a method?

Final methods cannot be overridden

64) What happens when a variable is marked as volatile?

A separate copy gets created for each and every thread

65) What is static variable?

A variable which is declared inside a class and declared with static modifier it is known as static variable.

66) Should default be the last case in a switch statement?

- no

67) What is an enhanced for loop?

For each loop or enhanced for loop is used to iterate the group of elements in an array or in a collection.

68) What are the default values in an array?

- For byte, short, int, long it is 0
- float, double it is 0.0
- char it is space
- boolean it is false
- String and any reference type it is null

69) How many ways are there to read input through the keyboard?

- BufferedReader and InputStreamReader classes
- Scanner class
- Console class
- CommandLine Arguments etc...

70) Difference between final, finally and finalize in java

- final is the keyword used to declare a variable value as constant. final variable values cannot be changed.
- finally is used in exception handling
- finalize method will be called before the object gets garbage collected. but it is deprecated now.

71) What are wrapper classes?

- Wrapper classes are mainly used to wrap the primitive content into an object.
- This operation of wrapping primitive content into an object is called boxing.
- The reverse process i.e., unwrapping the object into corresponding primitive data is called Unboxing.

72) Why do we need wrapper classes?

- int a=10;
- Integer I = a;//autoboxing
- int b = I;//auto unboxing

73) What is casting?

Converting from one datatype to another datatype is called casting.

74) What is String immutability?

String class are immutable in nature. That is ones we stored the data we cannot modify it. Even if we do modification that modification will not be done in same memory location.

75) Where are string values stored in memory?

If we create String class objects using new operator then memory will be created in Heap.

If we create String class objects using String literal then memory is allocated in String constant pool or String literal pool.

76) What is the difference between String and StringBuffer and String Builder classes

- String class objects are Immutable and Thread safe
- StringBuffer class objects are Mutable and Thread safe
- StringBuilder class objects are Mutable and non Thread safe.

77) How do u print contents of an array?

There are 3 ways to print contents or elements of an array.

- Using Arrays class toString()
- Using for loop
- Using for each loop

78) What are the different ways of creating Wrapper class instances?

- Using Boxing, and AutoBoxing

79) What is auto boxing

From JDK 1.5 onwards, Auto-Boxing is introduced.

According to this feature, you need not to explicitly wrap the primitive content into an object. Just assign primitive data to corresponding wrapper class reference variable, java automatically wraps primitive data into corresponding wrapper object.

```
byte b = 10; //Primitive byte data

Byte B = b; //Auto-Boxing of byte data
```

80) Can you give examples of different utility methods in String class?

charAt(), length(), substring(int index), startsWith(), endsWith(), indexOf(), lastIndexOf(), Contains(), format(), join(), split(), etc… refer java.lang.String class.

81) Why should you always use blocks around if statement?

Under if without curly braces we can write only single statement. But if we want to write multiple statements under if we should write curly braces.

82) How do you loop around an array using enhanced for loop?

```
for( datatype varname : array name / collection name)
    {
            //body of the loop
    }
```

83) What is an enum?

Enums in java are mainly used for grouping similar type of constants as a one unit. constants means static and final.

Enums are introduced in java in JDK 1.5 onwards

```
enum Directions

{

NORTH,SOUTH,EAST,WEST;

}

class EnumMainDemo1{

        public static void main(String args[]) {
        Directions d1 = Directions.EAST;
        Directions d2 = Directions.NORTH;
        System.out.println(d1);
        System.out.println(d2);
        System.out.println(Directions.SOUTH);
        }
        }
```

84) Can you use a switch statement around an enum?

- yes

85) What is Static Initialization block?

SYNTAX:

```
static

{

        //static block code goes here

}
```

- the main purpose of static blocks is to initialize static variables of the class.
- static blocks will be executed only ones when class is loaded into the memory.
- the purpose of constructor is to initialize the instance variables and it will be called by JVM each and every time whenever an object is created. Whereas static block is used for initializing static variables and it will be called by JVM only ones when class is loaded into memory.
- we should not initialize instance variables inside static blocks.
- If we use static block and main method in a program then JVM first executes static block and then main method.
- writing static blocks in a program is optional
- static blocks must be written inside a class but not inside a method.
- we can write multiple static blocks in a class but internally all of the static blocks will be treated as a single static block and the code inside that static blocks will be executed in the order how they are written in the program.
- when we define static variable initialization in our program without using static block then compilation environment will provide a system defined static block in that it will do that static variable initialization.

87. What is Instance Initialization block?

syntax:

```
{
//instance block code goes here
}
```

- The main purpose of instance block is to initialize instance variables of the class.
- its purpose is exactly similar to the constructor.
- instance block is executed each and every time when an object is created before executing constructor and after executing static block.
- we can write multiple instance blocks in a class but internally all of the instance blocks will be treated as a single instance block and the code inside that instance blocks will be executed in the order how they are written in the program.

- instance blocks must be written inside a class but not inside a method.

88. Difference between Static Method and Instance Method?

Instance methods: the methods which are defined inside a class are known as instance methods.

- Instance methods memory is created inside object in heap.
- we can use instance variables directly without using object in instance methods when both belongs to the same class.
- but when we want to use static variables in instance methods we should access by using class name or directly also.
- we can call one instance method from another instance method directly without using object when both belongs to same class.
- hen we call instance methods of one class in another class or in static methods then we should access by using object.

STATIC METHODS:

- when we define methods inside a class but declared with static keyword or modifier then those methods are called static methods.
- static methods memory is created in method area.
- only one copy of a static method is created for the whole class and that is shared by all objects of the class.
- static data is in the scope of class.
- we can use static variables and static methods directly from another static methods without using class name when all belongs to same class.

when we are accessing static methods from instance methods or from another class static methods then we should access by using class name.

89. What is the difference between this and super?

this and super are the default reference variables. this is always used to refer current class variables, methods.

super is used to refer super class variables, methods

90. What is the difference between this() and super()?

this() is used to call current class constructor explicitly from another constructor of same class. Or to achieve constructor chaining we use this() and super().

super() is used to call super class no argument constructor.

91. Can we use this() and super() at a time in a constructor?

No. we should not use because this() and super() should be the first statement to be executed in a constructor. So if we use both at a time we will get compile time error saying this() or super() must be the first statement to be executed error.

92. What is Constructor Overloading?

Defining more than one Constructor in a class but difference in parameters then those constructors are called overloaded constructors and concept is known as Constructor Overloading.

93. How can you achieve inheritance in Java? How many ways are there?

We can achieve inheritance using both extends and implements keywords.

- Single Inheritance, Multilevel Inheritance, Hierarchical Inheritance.

94) What are Object class methods?

Object class is the super class for all classes in Java.

getClass(), hashCode(), wait(), toString(), clone(), equals(), finalize(), notify(), notifyAll() methods

95) What is the default hashCode() implementation?

The default hashcode method in the object class returns a value a number that is nothing but the memory location of that particular object. So if we do not provide our own Hash code method, the value that will come back is the memory location of that object

96) What is the default toString() implementation?

In user defined classes if we try to print an object of that class, then we will see a value which will have the class name, followed by the @ symbol and the hexadecimal representation of the memory location of that particular object. So that is the default implementation in the object class.

The toString() method is implemented in such a way that it will always return the class name followed by @ symbol and then the hexadecimal representation of the objects address.

And if we override it, then we will see whatever implementation we provide, whatever we return back from the toString method that will be used instead.

97) Why are Strings Immutable?

String class objects are Immutable in nature. Ones we store the contents we cannot modify it. Even if we change also, modification will be done in another location.

98) Difference between Immutable and Mutable String objects?

If we want to create mutable String objects then we should use StringBuffer and StringBuilder classes. In this classes even if we do modification it will be done in same memory location. String class objects are immutable which cannot be modified. Modified string will be stored in another memory location.

99) Difference between equals() and == operator?

Both are used to Compare object references. But when we override equals() it compares the content.

100) Where are string values stored in memory?

If we create String using String literal, memory is allocated in String constant pool. But if we create String using new operator memory is allocated in Heap area.

101) What is the difference between String and StringBuffer and String Builder classes

All are final classes. String class Objects are Immutable and Thread safe

StringBuffer class objects are Mutable and Thread safe

StringBuilder class objects are Mutable and Non Thread safe

102) Explain about toString() method?

The toString() method is implemented in such a way that it will always return the class name followed by @ symbol and then the hexadecimal representation of the objects address.

And if we override it, then we will see whatever implementation we provide, whatever we return back from the toString method that will be used instead.

103) What is the use of equals method in java?

Equals method is used to compare two objects. This method is present in Object class. It is similar to that of == operator, used to compare references to objects. But if we want to compare contents present in objects then we should override this method in our class, then it compares the content of two objects. It has already overridden in String class, Wrapper classes, and Enum class.

104) What is the use of hashCode() method?

hashCode() method is used to get the unique reference of an object.

105) Can you give examples of different utility methods in String class?

length(), charAt(), contains(), substring(), startsWith(), endsWith(), indexOf() etc..

106) What are differences between String and StringBuffer?

String class is used to create Immutable objects. String class objects will be created by using String literal and new operator.

StringBuffer class is used to create Mutable objects. StringBuffer class objects are created by using new operator.

107) What are differences between StringBuilder and StringBuffer?

Both are used to create Mutable objects. But StringBuffer is threadsafe and StringBuilder is non thread safe.

108) Define Assert statements and when it is used?

An assertion allows testing the correctness of any assumptions that have been made in the program. An assertion is achieved using the **assert** statement in Java. While executing assertion, it is believed to be true. If it fails, JVM throws an error named **AssertionError.** It is mainly used for testing purposes during development.

The **assert** statement is used with a Boolean expression and can be written in **two different ways**.

First way: assert expression;
Second way: assert expression1: expression2;

109) What is Coupling?

Coupling is a measure of how much a class is dependent on other classes. There should be minimal dependencies between classes. So, We should always aim for low coupling between classes.

110) What is Cohesion?

Cohesion is a measure of how related the responsibilities(methods) of a class are. A class must be highly cohesive that is its responsibilities should be highly related to one another.

EXCEPTION HANDLING:

111) How can you handle exceptions in Java?

By using try catch and finally blocks.

112) Why is exception handling important?

By using Exception handling mechanism we can avoid the damage that should be happened for our data. We can provide our own user friendly exception messages.

113) What is the super class for all exceptions and errors in java?

Throwable class which is in java.lang package

114) What is the super class for all exceptions in java?

Exception class

115) What are the keywords used to handle an exception in java?

Try, catch, and finally keywords

116) Name different type of exceptions in java?

Checked and Unchecked exceptions

117) Can we write statements between try block and catch block?

No

118) What design pattern is used to implement exception handling features in most languages?

Chain of responsibility design pattern → a way of passing a request between chain of objects.

119) What is the need for finally block?

Clean up or closing operations will be performed in finally block. Finally block always gets executed where exception is there or not.

120) In what scenarios is code in finally not executed?

If JVM crashes in between like if we use System.exit() and if exception is thrown in finally.

121) Is try without a catch is allowed?

Yes. We can use try and finally blocks without a catch block.

122) What is exception handler in java?

The code that catches the exception thrown by JVM.

123) Can we just use try instead of finally and catch blocks?

No.

124) Can you explain the hierarchy of exception handling classes?

Throwable class is the super class for all exceptions and errors

Error class is the super class for all Error classes

Exception class is the direct super class for all checked exceptions(exceptions which are detected by compiler at compile time)

RuntimeException class is the super class for all unchecked exceptions(exceptions which are detected by JVM at runtime)

125) Difference between 1. System.out.println(1/0); 2. System.out.println(2.0/0);

In first case, Arithmetic Exception raises at run time

In second case, Infinity is the output.

126) What is the difference between error and exception?

Error is used in situations when there is nothing a programmer can do about an error.

Ex: VirtualMachineError, StackOverFlowError, OutOfMemoryError

Exception is used in situations when a programmer can handle the exception.

127) What is the difference between checked exceptions and unchecked exceptions?

Checked exceptions are detected by compiler at compile time. Exception class is super class for all checked exceptions.

Unchecked exceptions are detected by JVM at runtime. RuntimeException class is the super class for all unchecked exceptions. Exception class is super class for RuntimeException class.

128) Which Exception class can you use in the catch block to handle both checked and unchecked exceptions?

Exception class

129) Difference between throw and throws keywords?

Throw is used to throw an exception explicitly by the programmer.

Throws is used when the programmer don't want to handle the exception.

130) How do you throw an exception from a method?

Using throw keyword

131) What happens when you throw a checked exception from a method?

It raises a compile time error.(unhandled exception type exception)

132) What are the options you have to eliminate compilation errors when handling checked exceptions?

Declaring that a method would throw an exception

Handling the checked exception with a try catch blocks

133) How do you create a custom exception?

Create a user defined class which extends Exception class or RuntimeException class.

134) How do you handle multiple exceptions in a single catch block?

Using Pipe operator

135) Can you explain about try with resources?

When the try block ends the resources are automatically released. We don not need to create a separate finally block.

```
try(BufferedReader br = new BufferedReader(new InputStreamReader(System.in))) {

                        // logic
                    }
                    catch(Exception e)
                    {
                        e.printStackTrace();

                    }
```

136) What is the need for threads in Java?

Threads allow java code to run in parallel.

We want to do more things in less time

137) How do you create a thread?

By extending Thread class

By implementing Runnable interface

138) How do you create a thread by extending thread class?

```
class UserDefinedThread extends Thread {

public void run( ) {

        // logic

}

}
```

139) How do you create a thread by implementing runnable interface?

```
class UserDefinedThread implements Runnable{

        public void run()
        {
            // logic
        }

}
```

140) How do you run a thread in Java?

By using Thread class start() method

141) What are the different states of a thread?

New, Runnable, Running, Blocked / Waiting, Treminated / Dead states

142) What is priority of a thread? How do you change the priority of a thread?

Thread priorities ranges from 1 to 10. Default thread priority is 5.

We can change the thread priority using setPriority() method.

143) What is Executor Service?

Java.util.concurrent.ExecutorService interface is a new way of executing tasks asynchronously in the background. It is similar to that of a Thread pool.

144) Explain different ways of creating executor services

There are 3 ways of creating executor services.

ExecutorService executorService1 = Executors.newSingleThreadExecutor();

ExecutorService executorService2 = Executors.newFixedThreadPool(10);

ExecutorService executorService1 = Executors.newScheduledThreadPool(10);

145) How do you check whether an executionservice task executed successfully?

We can use a Future interface to check the return value. Future interface get method will return null if the task finished successfully.

146) What is callable? How do you execute a callable from executionservice?

Runnable interface run method has a return type void. It cannot return any result from executing a task. A Callable interface class method has a return type. If you have multiple return values possible from a task, we can use the Callable interface.

```
Future future = executorService1.submit(new Callable(){

    public String call() throws Exception{

        return “hello”;
```

```
                }
        });
        S.o.pln(future.get());
```

147) Can you give an example of a synchronized block?

```
synchronized(this){
        // logic
}
```

148) Can a static method be synchronized?

Yes

149) What is the use of join method in threads?

We can maintain the order of execution of threads.

150) Describe a few other important methods in threads?

isDaemon(), setDaemon(),setName(), getName(), getId(), yield(), sleep(), interrupted() etc..

151) What is a deadlock?

Lets consider a situation where thread1 is waiting for thread2 and thread2 is waiting for thread1. This situation is called a Deadlock. In this situation, both threads would wait for one another for ever.

152) What are the important methods in Java for inter-thread communication?

wait(), notify(), notifyAll() methods

153) What is the use of wait method?

This method is present in Object class. This causes the thread to wait until it is notified.

154) What is the use of notify method?

This method is present in Object class. This causes the object to notify other waiting threads.

155) What is the use of notifyAll method?

If more than one thread is waiting for an object, we can notify all the threads by using **notifyAll** methods.