Java Questions 2020

1. **Why Java ?**
    - Java programming language mainly designed to develop internet applications by providing platform independency.
    - C,C++ **programming** languages supports developing only stand-alone application, it can only be executed in current system, cannot be executed from remote system via network call.
2. **Definition of Java**
    - Java is a very simple, high-level, secured, multithreaded, Platform independent, object oriented programming language.
    - It was developed by James Gosling in SUN Microsystems in 1990's for developing internet applications. It's first version is released in Jan 23.1996.
3. **What is the abbreviation of Java ?**
    - There is no abbreviation for Java. The Development team of java has chosen this name.
4. **Who developed Java ?**
    - James Gosling and he is a famous developer, best known as the father of the java programming language.
5. **What is a Platform ? - Diagram**
    - Platform is a hardware or software environment in which a program runs. For instance, computer platform is (OS + Hardware devices)
6. **Difference between Platform dependent and Platform Independent ?**
    - **Platform Dependent :** An application that is compiled in one operating system, if it is not run in different operating system then that application is called platform dependent application.
    - **Platform Independent :** If the application compiled code is able to run in different operating system then that application is called platform independent application.
7. **Why the name OAK renamed to Java ?**
    - They were unable to register this programming language with name OAK, because already some other product is registered with the same name.so they renamed to java.
8. **What is the main feature of a programming language to develop internet application ?**
    - Platform independency because an internet application must run in all Operating System's irrespective of where it is compiled.
9. **Why C,C++ programming languages cannot support internet application development ?**
    - Because they are platform dependent.

10. **Why C,C++ programming languages were developed as platform dependent, why not as platform independent ? -** Diagram
    - By the time C/C++ programming languages were developed there was no internet. Platform Independency feature is only required for developing internet application.
11. **Why C,C++ program compiled code cannot execute in all operating systems ?**
    - In these programming languages, the compiled code is Machine Language which understandable only by the current OS.
    - So This compiled code cannot be executed in different operating System.
12. **What are the Java File's Extensions ?**
    - In Java we have only two extension files
        i. .java extension file it is called source code which is developed by developer
        ii. .class extension file it is called compiled code contains bytecode which is generated by compiler from source code.
13. **What is Java Slogan ? - Diagram**
14. **Who develop JVM, OS Vendor vs SUN ?**
    - JVM is developed by SUN separately for every OS, not by OS Vendor.
15. **Is JVM available for all Operating Systems ?**
    - Yes, JVM is available for every OS separately.
16. **Is JVM installed automatically with OS ?**
    - No, it is not installed with OS. We must install it in our computer explicitly.
17. **What are the developer and Client Responsibilities ?**
    - Developer is responsible of developing, compiling and executing the java application, so in developer system we must install both Compiler & JVM
    - Whereas client is responsible of only executing that java application, hence in client computer just installing JVM is sufficient.

18. **Difference between JDK,JRE,JVM and JIT Compiler ?**
    - **JDK** : It provides development tools compiler + Runtime Environments JRE by Using JDK we can compile and execute new applications and also we can modify existing applications this is the point we can cover about JDK
    - **JRE** : It provides only Runtime Environments it does not provide any development tools we can only execute already developed applications we cannot develop new applications and also we cannot modify existing applications
    - **JVM** : Java Virtual Machine and java platform it is responsible to execute our java bytecode .It provides interpreter plus and responsible to covert source code to byte code.
    - **JIT** : It is responsible to help interpreter for executing java Bytecode fast basically JIT will improve the performance of Java program execution then how JIT will increase the performance of java program execution.

19. Different environments existed in real time project development ?

- **Development Environments** : here Developers will work to develop new programs. Hence we should install JDK in development environment.
- **Testing Environments** : Here testers will work to test the project, means they just execute the project. Hence JRE installation is enough.
- **Production Environments:** Here end-user will work to use the project, means they just execute the project to complete their transactions. Hence JRE installation is enough.

20. Difference between Major and Minor Version ?

- Major Version – it is the main version of the java software contains new features & enhancements of current release of that software.
- Minor Version – It is the sub version of the major version contain bugfixes.

21. Difference between Project and Product ?

- A Project is a software that is developed specific to one customer.
- A Product is also a software but is developed common for all companies, we can say it is a ready made software.

22. What is the significance of the number 2 in editions like J2SE,J2ME,J2EE ?

- Nothing, it just represents editions were introduced in Java 2 Version.

23. What is JAVA_HOME ?

- The Java software installed directory is called JAVA_HOME. We also follow this naming convention for other software's like Oracle installed directory is called ORACLE_HOME and Tomcat installed directory is called CATALINA_HOME.

24. Difference between Binary files and Library files in Java Software ?

- Binary files are command files means they have commands to be executed in sequence. Java Binary files have commands to compile and execute java Program.
- Library Files are program files means they have logic that is used to develop another applications. Java library files are used in developing new java applications & applets.

25. How can we compile and execute java program and from where ?

- Java Programs are compiled and executed using java Binary files javac and java. We call these two binary files as "tools" in short form. These two tools are used from command Prompt.

26. Where Java files must be stored, in Java software installed folder or in some other directory ?

- It is not recommended to store java Source files in java Software installed directory , because if we uninstall current java software for installing next version, all our java sources files are also deleted. So for Security reasons it is always recommended to store java sources files in another directory.

27. Is Javac and Java tools are available from outside java software installed directory ?

- No, these files are not available from other folders.

28. What are the basic programming elements in Java ?

- Package, Class, Interface, Enum, Variables, Methods

29. What is the mandatory basic programming element ?

- Class is mandatory , because it allows us to define methods with logic.

30. Why does a java Program start with a class ?

- The real world object can be represented only using class.
- We can define methods with logic only using class.

31. What are the essential statements of java Program ?

- Class Block : Because only class allows us to define method with logic.
- Main method : because it is initial point of class logic Execution

32. Is Main method mandatory for compilation or execution ?

- Only Required for execution not for compilation

33. What does the compiler do if the given java file is not found ?

- It throws CE:javac:n file not found:>filename>

34. What does JVM do if the given class's .class file is not found ?

- It Throws RE: java.lang.NoClassDefFoundError:<className>

35. Can we create empty java file, can we compile and execute it ?

- Yes But after compilation we done have .class file so we cannot execute.

36. Can we create empty class, can we compile and execute it ?

- Yes we can create and compile empty class. Compiler generates .class file with that class name. we cannot execute this class as it does not have main method. It leads to exception.

37. Is it mandatory that java file name and class name should be same ?

- No, not always- file name can be user defined name.

38. When should the Java file name and class name be the same ?

- If class is declared as public, file name should be the same as the public class name else its name can be user defined.

39. In a single java file how many classes can we define ?

- We can define more than one class. The rule is class name should be different.

40. If a Java file has multiple classes what is the java file name ?

- Public class Name , if there is no public class, java file name can be user defined.

41. In a Java file, how many public classes can we define ?

- Only one. We can define one public class and multiple non-public classes.

42. If we compile multiple classes java file, how many .class files are generated by compiler ?

- Compiler generates .class files as many class definitions as we have in that java file. Compiler generates .class file separately for every class with that class name.

43. What is meant by user defined method and predefined method ?

- Developer defined method is called user defined or custom method.
- Already defined methods are called predefined methods.
- These methods may be developed by SUN Microsystem developers or some other developers. Ex : println()

44. What is meant by user defined class and predefined class ?

- Developer defined class is called user defined or custom class.
- Already defined classes are called predefined classes.
- These methods may be developed by SUN Microsystem developers or some other developers. Ex : System , String

45. Does JVM execute user defined methods automatically ?

- No, it executes only main method. To execute user defined methods we must call them from main method or method calling from main.

46. Does JVM execute user defined methods by itself just by declaring them as static ?

- No, Static methods also should be called from main method.

47. Then How main method is executed by JVM Automatically ? Is it not because of static keyword ?

- Yes, Main method is executed not because of static keyword, because it is called from JVM Software.

48. Definition of main method

- Main method is the mediator method between java developer and JVM to inform what are the methods should be execution when and in which order.

49. How many user defined methods can we define in a class ?

- We can define multiple methods, but rule is method names should be different.

50. What is the order of execution of all user defined methods ?

- In the order they are called from main method or from any one of the method calling from main method.

51. Can we call main method explicitly ?

- Main(new String[0])

52. What does happen when we pass negative number as size to array Object ?

- Main(new String[-4]) : it leads exception – NegativeArraySizeException

53. What does happen when we call main method in the same class ?

- No CE, it leads exception : StackOverflowError because it leads to recursive method call.

54. Can we overload main method ?

- Yes

55. What is the difference between System.out.print() and System.out.println()?

- Println() method places the cursor in the next line after printing current output. So that the next coming output will be printed in next line.
- But whereas print() method places the cursor in the same line after printing the current output. So that the next coming output will be printed in same line.

56. Why pointers are eliminated from Java ?

- Pointers lead to confusion for a programmer.
- Pointers may crash a program easily, for Example : when we add two pointers ,the program crashes immediately.

- Pointers break security, Using pointers harmful programs like virsus and other hacking programs can be developed. Because of the above reasons, pointers have been eliminated from Java.

57. What is JIT Compiler ?

- JIT Compiler stands for just-in-time compiler. It is the part of JVM which increases the speed of execution of a java program.

58. Why Comment ?

- Comments are meant for developers to understand the purpose.

59. How many types of Comments in Java ?

- Single Line Comment - //
- Multiline comment - /* */
- Document comment - /** */

60. What is an Identifier and what are the rules need to follow ?

- Any name in the java program like variable name , class name , method name , interface name is called Identifier.
- Identifier is a name of the basic programming elements.
- Rules :
  i. Identifier should contain only alphabets, Digits and Special Characters
  ii. Identifier should not start with any number, it may start with alphabet symbol and underscore symbol and dollar symbol.
  iii. Identifier should not be duplicated
  iv. It is possible to declare predefined class names and predefined interfaces name as identifier but it is not recommended to use.
  v. Identifier should not contain space between words. If you want to give gap then use underscore symbol between two words.
  vi. Keywords cannot be used as user defined identifier.

61. Is null a keyword ?

- No, it is not a keyword. It is a literal.

62. Can we use null as user defined identifier ?

- No, even though it is not a keyword we cannot use it as identifier.

63. How many datatype keywords java supports ?

- 8

64. How many types of datatypes java supports ?

- Primitive types – used to store single value at a time
- Referenced types – used to collect multiple of values using primitive types

65. How many modifiers java supports ?

- 11

66. How many accessibility levels java suppors ?

- Four, among them 3 are keywords.

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

- Final : is a keyword used to create constant variable, method and class
- Finally : is a keyword used to define a block of statements to be executed definitely for a try block.
- Finalize : is a method used to define logic to be executed definitely before an object is destroyed.

68. Why do we have 8 primitive types in Java ?

- Based on type and range of data, primitive types are divided into 8 types.

69. Why do we have 4 referenced types ?

- To collect same type of values Array is given.
- To Collect different type of values class, interface, enum are given.

70. What is the need of datatype ?

- Data types are used to store data temporarily in computer through a program.

Static Members and Their Control Flow

1. **What members are called static members ?**
    - The class level members which have static keyword in their definition are called static members.
2. **How many types of static members ?**
    - Static Variables
    - Static Blocks
    - Static Methods
    - Main Method
3. **When do all static members get memory location and by whom ?**
    - All static members are identified and get memory location at the time of class loading by default by JVM in method area.
4. **What is a static variable ?**
    - A class level variable which has static keyword in its creation statement is called static variable.
5. **Does JVM execute static variables by default ?**
    - Yes, it executes static variables means provides memory location at the time of class loading.
6. **How many static variables can we define in a class ?**
    - Multiple static variables we can define but every variable name should be different.
7. **What is the order of execution of all static variables ?**
    - In the order they are defined from top to bottom
8. **When, where, how and by whom memory location is providing to static variables ?**
    - When class is loaded JVM provides individual single copy of memory location to each static variable in method area only once in a class life time.
9. **What is the lifetime and scope of static variable ?**
    - Static Variable get life as soon as class is loaded into JVM and is available till class is removed from JVM or JVM is shutdown.
    - Class scope means it is accessible through the class directly by its name and outside class by its class name provided it is non-private.
10. **Can we declare local variables or parameters as static ?**
    - No, it is not possible it leads to CE: illegal start of expression
    - Static keyword is not allowed inside a block {} because static variable must get memory at the time of class loading which is not possible to provide memory to local variable at the time of class loading.
11. **Can we create local variable with parameter name ?**
    - No because both parameter and local variables are created in the same method scope so local variable is considered as duplicate variable.
12. **What is Shadowing ?**
    - Creating a local variable or parameter with same static or non-static variable name is called shadowing. It means local variable is a shadow of class level variable.

13. How static variable can be differenced from local variable or parameter when both have same name ?

- We should use class name to differentiate static variable from local variable.

14. What is a static Method ?

- A method which has static keyword in its definition is called static method.

15. Does JVM Execute static method by default like static variables ?

- No, JVM does not execute static methods by itself. They are execute only if they are called explicitly by developer either from main method or from static variable as its assignment statement or from static block.

16. What is the order of execution of static methods ?

- In the order they are called, not in the order they are defined.

17. Where static methods logic is stored and where they are executed ? - JVM

- All static methods logic is stored in method area and that logic is executed in java stacks area in main thread by creating separate stack frame.

18. What is a static blocks ?

- Static Block is a class level nameless block that contains only static keyword in its prototype.

19. What is need of static block?

- It is used to execute logic only at the time of class loading.
- Logic like,
- Initializing static variables
- Registering native libraries
- To know classes loading order, etc…..

20. Who will execute static block when and where?

- Static blocks are executed automatically by JVM at the time of class loading in the order they defined from top to bottom by creating separate stack frame in main thread in Java stacks area.

21. Then what is order of execution of SB and Main method?

- Static block is always executed before Main method.

22. Can we execute class logic without main method?

- Yes, we can execute by using either static variables or static block, but from Java 7 main method is mandatory to execute a class.

23. Class execution procedure with static block and main method?

- When a class is loaded, first JVM executes static variables and static block then after it searches from main method, if main method presents is execute main method else it terminates program execution with exception java.lang.NosuchMethodError.
- But from Java 7 onwards class execution starts only if it has main method, else program execution is terminated without executing static variables or static blocks by throwing an
    i. ERROR: main method is not available

Main Method Interview Questions

1. **Can we execute main method at the time of class loading?**
    - Yes, it is possible. Call it from static block it is also executed at the time of class loading.
2. **Why main method is public?**
    - Because it must be called by JVM from out side our package.
3. **Why main method has a static keyword in its definition ?**
    - Main() is the initial point of class logic execution. Hence it should be identified at the time of class loading.
    - Due to this reason it contains static keyword in its prototype.
    - Of course it must be executed without object creation.
4. **Why JVM Executes only main method why not user defined static methods ?**
    - Main Method calling statement is available in JVM Software but our static methods calling statement is not available in JVM Software.
5. **Definition of Main Method ?**
    - Main method is the mediator between java developer and JVM to inform methods execution order.
6. **Why main() method return type is void ?**
    - Because if we return a value, it is sent to JVM which is useless. Hence main() method return type is void.
7. **Why its name is main ?**
    - As per the coding standards the method name should convey the operations it is doing. Since this method is intended to execute main logic of the program, that is business logic, it is name as main.
8. **Why main() method has parameter String[] ?**
    - To read command line arguments(values) in to java application from keyboard.
    - Java Sample 10 20
9. **Why main method parameter type is String ?**
    - Because the value passing from keyboard is sent to java application as String type value.
10. **Why is it String[], why not it is just String ?**
    - To read more than one value
11. **Why main() method parameter name is args ?**
    - Again it is also because of coding standard as we reading arguments from keyboard the name of the parameter is “args”. We can change its name, but it is not recommended.
12. **Can we call main method explicitly ?**
    - Yes it is possible, it is also a method.
    - Note : main() method should not be called from its own block or from a method that is calling from main(), it leads to exception java.lang.StackOverflowError.

Non-Static Members and their Control flow

1. **What members are called non-static members ?**
    - Class level members which don't have static keyword in their creation statement are called non=static members.
2. **What are the types of non-static members ?**
    - Non-static Variables
    - Non-static Blocks
    - Non-static methods
    - Constructors.
3. **When do all these members get memory location and by whom ?**
    - All above members get memory location only if object is created with new keyboard and constructor of that class.
    - JVM will not provide memory location for these members by default by itself.
4. **What is an object and its creation of syntax ?**
    - Object is an instance of a class that contains continuous memory location of all non-static variables of a class with specific data of this instance.
    - Object can be created by using new keyword and constructor of that class.
5. **What is non-static variable ?**
    - The class level variable that does not have static keyword in its definition is called non-static variable
6. **Will JVM executes non-static variables automatically ?**
    - No, JVM executes non-static variables only if object is created.
7. **When, where, how and by whom non-static variables get memory location ?**
    - Non-static variables get memory location in heap area in continuous memory locations by JVM when object is created.
8. **How many objects can be created for a class ?**
    - We can create multiple objects for a class but their referenced variable names must be different.
    - When we create multiple objects, non-static variables get separate copy of memory for each object. So to access non-static variables from a particular object we must use that object's referenced variable.
9. **If we modify one object data will another object data also be modified ?**
    - No, Modifications done for one object will not be affected to another object, because we change object data using its referenced variable.
10. **What is null referenced variable ?**
    - If we store null in referenced variable, it is called null referenced variable . it will not point to any non – static variable or methods of the class.
11. **What is object referenced variable ?**
    - If we store object reference in referenced variable, it is called object referenced variable. It points to all non-static variables or method of the class.
12. **What is the output printed if we print referenced variable/object ?**
    - If we print null referenced variable, print() and println() methods prints →null
    - If we print object referenced variable, print() and println() methods prints → classname@hascode in hexadecimal string format.

13. When you print object, how it's hash code is printed in its hexa String format ?

- It is the internal implementation of toString() method.

14. What is toString() method ?

- It is a predefined method available in java.lang.Object class to return object information in string format.
- It's default implementation is returning object [cassname@hashcode in hexadecimal string format]

15. What is the method called internally by print() and println() methods to print object ?

- toString() it will print object information if you original object values then need to override with all variables.

16. What is the output printed if we print null referenced variable ?

- Employee emp = null;
- System.out.println(emp)
    - To print null referenced variable, print() and println() methods does not call toString() method because it leads to java.lang.NullPointerException.

17. Can we access static variables or static methods using referenced variable ?

- Yes, we can access because they get memory directly at the time of class loading and more over they are part of object.

18. Can we access non-static variable or non-static method with null referenced variable ?

- No, it leads to RE:java.lang.NullPointerExcepton because non-static variables or non-static methods does not have memory with respect to null referenced variable.

19. Can we access static variable or static method with null referenced variable ?

- Yes , there is NO CE, and NO RE, program is executed successfully, because static variable or static method gets memory directly with respect to class.

20. What are the different ways to access static and non-static variables, and what is the effect in modifying static and non-static variables with an object ?

- Static Variables can accessed using
    - It's name directly
    - Class name
    - Null referenced variable
    - Object Referenced variable name – means object
- Non-Static variables can accessed using
    - Object reference variable name

21. What is the importance of Non-static variable ?

- If non-static variables are accessed directly by their name or using class name it leads to compile-time error "non-static variable cannot be referenced from static context"
- If non-static variables are accessed using null referenced variable it leads to run time error "java.lang.NullPointerException"

22. What is non-static methods ?

- Methods which doesn't have static keyword in its prototype is called non-static method.

23. Does JVM execute non-static method automatically when object is created ?

- No, we should call them explicitly using object. If you call them directly from main method it leads to CE: “non-static method cannot be referenced from static context”

24. Can we create object without having non-static variable in the class ?

- Yes it is possible. JVM creates dummy object.
- Dummy object means it does not have any state, but will have behaviour and hashcode.

25. What is the order of execution of non-static methods ?

- In the order they are called.

26. Where non-static methods are stored and executed ?

- Non-static methods logic(bytecodes) is stored in Method Area, and when it is called its logic is loaded and executed in Java Stacks area by creating separate stack frame in main thread.

27. How can we call non-static members from other non-static members ?

- Directly by their name.

28. How is it possible to call non-static members directly without object ?

- We can call non-static members from other non-static members directly, since we can guarantee providing memory for non-static variables and non-static methods before their calling statement is executed by JVM.

29. What is a current object ?

- The object that is used to call a non-static method is called current object.

30. Who will place this keyword in calling non-static members from other non-static members ?

- In compilation phase compiler places this keyword in all non-static members call if they are called directly by their name.

31. Can we create object in non-static method ?

- Yes, we can also create object in non-static methods, this object is called local object.

32. Can we pass object as an argument to a method ?

- Yes we can pass object as an argument to a method.

33. What is shadowing – local Preference in non-static members ?

- Creating a local variable with the same non-static variable name is called Shadowing.
- In a non-static method if there is any local variable or parameter defined with same non-static variable name always local variable values is retrieved.

34. When does compiler place this keyword in accessing non-static variables or when developer should place this keyword explicitly in accessing non-static variables ?

- If there is any local variable or parameter defined in the current method with the same non-static variable name compiler does not place this keyword.
- In this case developer must us this keyword explicitly to access non-static variables.

Constructor Interview Questions

1. **What is a constructor ?**
    - Constructor is a special method given in OOP language for creating and initializing object.
    - In Java, constructor role is only initializing object and new keyword role is creating object. In C++ constructor alone creates and initializes object.
2. **What are the rules need to follow while defining constructor ?**
    - Constructor name should be same as class name.
    - It should not contain return type.
    - It Should not contain modifiers.
    - In Its logic return statement with value is not allowed.
    - Constructor must be called along with new keyword else it leads to compile time error.
    - All Java legal statements are allowed except return statement with value.
3. **Can we define a method with same class name ?**
    - Yes it is allowed to define a method with same class name.
    - No Compile time error and no runtime error is raised but it is not recommended as per coding standards.
4. **If we place return type in constructor prototype will leads to CE?**
    - No, because compiler and JVM consider it as method.
5. **How compiler and JVM can differentiate constructor and method definitions if both have same class name?**
    - By using return type, if there is a return type is considered as method else it is considered as constructor.
6. **How compiler and JVM can differentiate constructor and method invocations if both have same class name?**
    - BY using new keyword, if new keyword is used in calling then constructor is executed else method is executed.
7. **Why return type is not allowed for constructor?**
    - As there is a possibility to define method with same class name, return type is not allowed to constructor to differentiate constructor block from method block.
8. **Why constructor name should be same as class name?**
    - Every class object is created by using the same new keyword, so it must have information about the class to which it must create object. For this reason constructor name should be same as class name.
9. **Can we declare constructor as private?**
    - Yes, we can declare constructor as private. All four accessibility modifiers are allowed to constructor.
    - We should declare constructor as private so not to allow user to create object from outside of our class. Basically we will declare constructor as private to implement singleton design pattern.
10. **Is constructor definition mandatory in class?**
    - No, it is optional. If we do not define constructor complier define constructor.

11. Can we define empty class? If so, is it really an empty class?

- Yes we can define empty class, but after compilation it has constructor definition that is placed by complier. So really it is not an empty class.

12. How can we know that compiler places constructor in class file?

- We should use "javap" tool. It is a java binary file available in "jdk\bin" folder. It is used to decompile class file to check that class members.

13. How many types of Constructor ?

- Default Constructor
- No-Argument | Non-Parameterized constructor
- Parameterized Constructor

14. What is Default Constructor , Non-Parameterized Constructor & Parameterized Constructor?

- Compiler given constructor is called default constructor. It does not parameters and logic except super() call
- Developer given constructor without parameters is called no-arg | non-parameterized constructor.
- Developer given constructor with parameters is called parameterized constructor.

15. What is the optional rule while defining constructor ?

- Super() call must also be placed as a first statement in developer given constructors. If developer does not place it, compiler places this statement

16. Why compiler given constructor is called default constructor ?

- Because it obtain all its default properties from its class
  i. Its accessibility modifier is same as its class accessibility modifier.
  ii. Its name is same as its class name.
  iii. It does not have parameters and logic.

17. What is the default accessibility modifier of a default constructor ?

- It is assigned from its class. So it may be default or public.

18. Why compiler defines default constructor without logic and parameters ?

- Because compiler does not know logic required for your class. The Object's initialization logic is different from one class to another class.
- **Note** : It places super() method call in all constructors because it is generic logic required for every class for calling super class constructor in order to initialize super class non-static variables when subclass object is created.

19. When does compiler provider default constructor ?

- Only if there is no explicit constructor defined by developer.

20. When developer must provide constructor explicitly ?

- If we want to execute some logic at the time of object creation that logic may be object initialization logic or some other useful logic.

21. If class has explicit constructor , does it has default constructor ?

- No. Compiler places default constructor only if there is no explicit developer given constructor.

22. Can we consider both default and no-arg constructor are same ?

- No, both are different. They are seems to be same, but really not same.

23. Difference between no-argument and default constructor ?

Default Constructor	No-arg constructor
It is given by compiler	It is given by developer
It accessibility modifier is same as class accessibility modifier. So the only allowed accessibility modifiers are default or public	It can have all 4 accessibility modifiers as it is defined by developer. So the allowed accessibility modifier are private, default, protected and public.
It does not have logic except super() call	It can have logic including super() call.

24. Can we define a class with public accessibility modifier and a constructor with other accessibility modifier ?

- Yes we can define

25. When should we define parameterized constructor in a class ?

- To initialize object dynamically with user given values then we should defined parameterized constructor.

26. What is constructor overloading ?

- Defining multiple constructors with different parameter types | order | list is called constructor overloading.

Non-Static Block Concepts

1. **What is a non-static block ?**
   - A class level block which doesn't have prototype(head) is called non-static block.
2. **What is the need of non-static block ?**
   - We should define non-static block to execute some logic only at the time of object creation irrespective of the constructor used in object creation.
3. **Who does execute NSB when and where ?**
   - NSB is executed automatically by JVM for every object creation in Java Stack area by creating separate stack frame in main thread.
4. **How many non-static blocks can be defined in a class ?**
   - We can define multiple non-static blocks.
5. **What is the order of execution of all NSB's ?**
   - All non-static blocks are executed by default by JVM in the order they defined from top to bottom before the invoked constructor.
6. **What is the order of execution of all non-static members ?**
   - When an object is created first all non-static variables and non-static blocks are executed in the order they are defined from top to bottom, then the invoked constructor logic is executed.
   - Non-static methods are executed only if they are invoked from any of the static or non-static members.
7. **What JVM does internally when an object is created ?**
   - When we create object first control is sent to invoked constructor, but its logic will not be executed instead non-static variables and non-static blocks are executed first in the order they are defined from top to bottom, then after constructor logic is executed.
   - If all non-static variables and non-static blocks execution is completed, we can say object creation is completed.
   - If constructor execution is completed, we can say object initialization is completed.
8. **Can we create object in static and non-static blocks ?**
   - Yes we create object in both static and non-static blocks. But due to the object creation in non-static block it leads to **StackOverflowError**, because constructor and non-static blocks are called recursively.
9. **Can we create object at class level using static and non-static referenced variables ?**
   - Yes, we can create object at class level using static and non-static referenced variables but the object creation using non-static referenced variable also leads to StackOverflowError.
10. **Important Note for accessing variables**
    - A class level variable should not be accessed directly before its creation statement either from other variables or from blocks it leads to CE: illegal forward reference.
    - But it is possible to access those variables from methods and constructors because they are always executed after variables execution.
    - Static Variable with "class Name" and Non-static variable with "this"

11. Different ways to execute static & non-static members from other class members ?

- Static members can be accessed in the below 3 ways
    i. Using it class name : Example.m1()
    ii. Using null referenced variable
        1. Example emp = null
        2. emp.m1()
    iii. Using object or object name or object referenced variable
        1. Example e = new Example()
        2. E.m1()
- Non-static members can be accessed in 1 way
    i. object name or object referenced variable

Polymorphism

1. **What is Polymorphism ?**
    - Providing multiple implementations to a method is called Polymorphism.
2. **How many times of polymorphism ?**
    - Compile Time polymorphism
    - Run-time Polymorphism
3. **What is Method Hiding ?**
    - Redefining super class static method in subclass with same prototype is called Method Hiding
4. **What is Method Overriding ?**
    - Redefining super class non-static method in subclass with same prototype is called Method Overriding
5. **What is Method Overloading ?**
    - Defining new method with the existed method Name but different parameters type | list | order is called Method Overloading.
6. **How many times a superclass is loaded into JVM ?**
    - A class is loaded into JVM only once. So a superclass is loaded into JVM only once when its first subclass is loaded into JVM.
    - When we load its next subclass superclass is not loaded again, only the current subclass is loaded.
7. **When a method must be overridden ?**
    - If the super class method logic is not fulfilling subclass business requirements, sub class should override that method with required business logic.
    - Usually in super class, methods are defined with generic logic which is common for all subclass.
8. **When a subclass method is treated as overriding method ?**
    - If a method in sub class contains signature same as super class non private method, sub class method is treated as overriding method and super class method is treated as overridden method.
9. **What is overriding rules in Java ?**
    - Overriding and access modifiers
        i. Protected-> public, protected
        ii. Default->default, protected, public
        iii. Public ->public
    - Final Methods can't overridden
    - Static Method can't overridden
    - Private methods can't overridden
10. **What are the exception rules need to follow while defining method ?**
    - If the superclass method does not declare an exception, subclass overridden method cannot declare the checked exception but it can declare unchecked exception.

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

11. How can we execute super class method if it is overridden in sub class ?

- Using super keyword.

12. How can we execute super class overridden static method from subclass ?

- By using super class name.

13. Can we override main method in subclass, if so are both methods executed ?

- We can override main method in subclass, and it is only executed from sub class. To execute super class main method, we must call it explicitly from sub class main method.

14. When should we overload methods ?

- To execute same logic with different type of arguments we should overload methods.

15. Can we overload methods in same class ?

- Yes it is possible No CE and No RE. Methods can be overloaded in same or in super and sub classes because overloaded methods are different methods but we cannot override method in same class it leads to CE " Method is already defined".

16. What should be the method parameter to accept all types of objects as arguments.

- Java.lang.Object class

Abstract class & Interface

1. **What is abstraction ?**
    - Abstraction is means hiding implementation details ,showing functionality to the user is called abstraction.
2. **How many ways to implement abstraction ?**
    - Interface
    - Abstract class
3. **What is abstract method ?**
    - A method does not have body is called abstract Method
4. **Why abstract class cannot be instantiated ?**
    - Because it is not fully implemented class so its abstract method cannot be executed.
    - If compiler allows us to create object for abstract class, we can invoke abstract method using that object which cannot be execute by JVM at runtime.
    - Hence to restrict calling abstract methods compiler does not allow us to instantiate abstract class.
5. **Who will provide implementation for abstract methods ?**
    - Sub class developers provide implementation for abstract methods according to their business requirements.
    - Basically in projects abstract methods are defined by super class developer and they are implemented by sub class developer.
6. **What type of members we can define in an abstract class ?**
    - We can define all static and non-static members including constructors plus abstract method. So in an abstract class we can define 9 types of members.
7. **Will abstract class memory be created when subclass object is created ?**
    - Yes, its non-static members get memory when its concreate subclass object is created.
8. **How can we execute static and non-static concrete members of abstract class ?**
    - Static members can be executed directly from its main method and its non-static members are executed by using its concrete sub class object.
9. **Can we declare abstract method as static ?**
    - No, we are not allowed to declare abstract method as static. It leads to compile time error
    - Illegal combination of modifiers abstract and static.
    - If compiler allows us to declare it as static, it can be invoked directly which cannot be executed by JVM at runtime.
10. **Can we declare abstract methods as final ?**
    - No, because it should be allowed to overridden in sub class. It leads to CE
11. **Can we declare abstract method as private ?**
    - No, because it should be inherited to sub class. It leads to CE
12. **What are the legal modifiers allowed in combination with abstract modifier ?**
    - Only two modifiers are allowed with abstract
        i. Protected
        ii. Public

13. Can we declare concrete class as abstract ?

- Yes it is allowed. Defining a class as abstract is way of preventing someone from instantiating a class that is supposed to be extended first.

14. How can you restrict subclass not to override super class method ?

- Final

15. How can you force subclass to override super class method ?

- By creating method as abstract method.

16. What is interface in Java ?

- Interface is a blue print of a class that have static constants and abstract methods.
- It can be used to achieve fully abstraction and multiple inheritance.
- It is used to achieve loose coupling.

17. Can there be any abstract method without abstract class ?

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

18. Can we define private and protected modifiers for variables in interfaces ?

- No, they are implicitly public, static final

19. What is the need of interface when we have abstract class to define abstract methods ?

- Java does not support multiple inheritance with classes.
- So we must use interface as super class to develop abstraction for supporting multiple inheritance.
- If we define abstract class in place of interface, a service provider cannot implement multiple specifications so that service provider cannot have multiple businesses.

20. Can we apply abstract keyword to interface ?

- Yes, it is optional and at compilation time it is removed by compiler

21. Can we create empty interface ?

- Yes, It is possible.

22. How can we write logic in interface ?

- Using inner class we can provide logic in interface

23. Main differences between interface & abstract class ?

- Interface is a fully unimplemented class used for declaring operations of a object.
- Abstract class is a partially implemented class. It implements some of the operations of the object those are declared in its interface.
- These implemented operations are common for all next level subclasses. Remaining operations are implemented by the next level subclasses according to their requirement.
- Interface allows us to develop multiple inheritance so we must start object design with interface, where as abstract class does not support multiple inheritance so it always comes next to interface in object development graph.

24. What are the similarities between interface & abstract class ?

- Both interface and abstract class cannot be instantiated but abstract class can be instantiated indirectly via sub class.
- Reference variable can be created for both interface and abstract class.
- Sub class should implement all abstract methods.
- Both cannot be declared as final.

25. Difference between abstract class vs Interface ?

Interface	**Abstract class**
1)Interface contain only abstract methods, default methods & static methods	1)Abstract class contain abstract and non-abstract methods
2)Interface supports multiple inheritance	2)It doesn't support multiple inheritance
3)Interface has only static and final variables	3)abstract class contain static, non-static and final variables
4)Interface can't main method and constructor	4)abstract class can have static methods, main method and constructor
5)Interface can't provide implementation of abstract class	5)Abstract class can provide the implementation of interfaces
6)interface keyword Is used to define interface	6)abstract keyword is used to declare abstract class interface

Array Interview Questions

1. **What is an array ?**
   - Array is a referenced data type used to create fixed number of multiple variables of same type to store multiple values of similar type in continuous memory locations with single variable name.
   - It create continuous memory locations using primitive types or reference types.
2. **What is need of array ?**
   - In project array is used to collect/group similar type of values or objects to send all those multiple values with single call from one method to another method either as an argument or as a return value.
3. **What is array limitation ?**
   - Its size is fixed, means we cannot increase or decrease its size after its creation.
4. **Find out valid array declaration statements ?**
   - **int**[] i;
   - **int** []j;
   - **int** [] k;
   - **int** l[];
   - **int**[5] m;
   - []**int** n;
5. **How many string objects are created from the below statement ?**
   String[] array =new String[5]
   - Zero, String Objects are created. It creates one string array object with 5 variables of type string with default value “null”
6. **What is the output from the below statement ?**
   Thread[] th = new Thread[‘a’] -> sop(th.length)
   - 97
7. **How many thread objects are created from above program ?**
   - Zero thread objects are created. Only one thread array object is created with 97 locations.
8. **If we declare array as final, will all its locations are also final ?**
   final int[] ia2 = new int[5]
   - No, only array object referenced variable is final. It means in the above example only “ia2” is final not its array locations.
   - It means we can modify array locations value but we cannot assign new array object reference to this final referenced variable. It leads CE.
9. **Can we declare array locations as final ?**
   - No, because we are not creating array locations.
10. **Can we declare a class referenced variable as final ?**
    - Yes, in this case also only referenced variable is final but not its objects variables.

11. Can we declare a class object's variables as final, in the above x,y ?

- Yes, it is possible because those variables are created by us. We must declare them as final in the class definition.

12. What are the advantages and disadvantages of Array ?

Advantages:

- We can put in place other data structures like stacks, queues, linked lists, trees, graphs, etc. in Array.
- Arrays can sort multiple elements at a time.
- We can access an element of Array by using an index.

Disadvantages:

- We have to declare Size of an array in advance. However, we may not know what size we need at the time of array declaration.
- The array is static structure. It means array size is always fixed, so we cannot increase or decrease memory allocation.

13. Can we change the size of an array at run time?

- No we cannot change the array size. Though there are similar data types available which allow a change in size.

14. Can you declare an array without assigning the size of an array ?

- No we **cannot** declare an array without assigning size.

15. What is the default value of Array ?

- Any new array is always initialized with a default value as follows
- byte, short, int, long – default value is zero (0).
- For float, double – default value is 0.0.
- For Boolean – default value is false.
- For object – default value is null.

16. How to compare two arrays ?

- If 2 arrays are of the same size & data type then comparison can be done using "Arrays.equal ()"

17. How to sort an array ?

- Arrays.sort()

18. Can we declare array size as a negative number ?

- No, We cannot declare the negative integer as an array size.
- If we declare, there will be no CE (NegativeArraySizeException)

19. When will we get ArrayStoreException ?

- It is a runtime exception.
- We can store only string elements in a string array. If anybody tries to insert integer element in this String array then we will get ArrayStoreException at run time.

20. Can we add or delete an element after assigning an array ?

- No it is not possible.

21. What is the meaning of anonymous array ?

- Anonymous array means array without any reference.
- Sop(new int[]{11,12,13,14,15}.length)

22. Is there any difference between int[] a and int a[] ?

- No, difference both are legal statement.

23. Int a[] = new int[3]{1,2,3} – is it right way to declare arrays in java ?

- No . We should not mention the size of the array when we are providing the array elements.

24. How to copy an array into another array ?

- Using For loop
- Arrays.copyOf() method
- System.arraycopy()
- Clone method

25. How do we search a specific element in an array ?

- We can use Arrays.binarySearch() method. This method uses binary search algorithm.

26. Where is the memory allocated for arrays in Java ?

- In Java, memory for arrays is always allocated on the heap as arrays in Java are objects.

String Interview Questions

1. **What is a String ?**
    - String is a sequence of characters placed in double quotes “”.
    - All string variables are instances of the String class.
    - String is class in java and defined in java.lang package.
    - String is immutable and final in java and JVM used String pool to store all the string objects.
    - It’s not a primitive data type like int and long.
2. **What are the possible ways to create String Object ?**
    - By assigning string literal directly
        - **String str =”abc”**
    - By using new keyword
        - String s = new String(“java”)
3. **Why String object is immutable ?**
    - Because of String pooling concept.
    - It increases **security** because any hacker can’t change its value and it’s used for storing sensitive information such as username and password.
        - you can open any file in java by passing the name of the file as argument to File I/o classes.
    - Ex: suppose 5 reference variables ,all refers to one object “Sachin”.
        - If one reference variable changes the value of the object, it will be affected to all the reference variables. That is why string is immutable in java.
4. **What is the advantage of Immutable ?**
    - **Security** : **Java class loading mechanism works on class names passed as parameters, then these** classes are searched in class path. Imagine for a minute, Strings were mutable, then anybody could have injected its own class-loading mechanism with very little effort and destroyed or hacked in any application in a minute.
    - **Performance**
    - **Thread Safe**
    - **Cache support :** Frequently we used hashcode of string in Hashset and HashMap.
5. **Why String Pooling ?**
    - To improve performance by saving memory.
6. **What is String Pool ?**
    - String pool is contain pool of strings stored in Java heap memory.
    - String pool helps saving lot of memory and increase performance.
    - When we use double quotes to create a string, it first looks for string with same value in the string pool. if found it just returns the reference else it creates a new string in the pool and returns the reference.
7. **What is StringBuffer class ?**

- It is a thread-safe, mutable sequence of characters.
- A String Buffer is like a String, but can be modified in the same memory location.

8. What is StringBuilder class ?

- StringBuilder class functionality is exactly same as StringBuffer class.
- StringBuffer is synchronized where as StringBuilder is not synchronized.

9. What are the difference between String, StringBuffer and StringBuilder ?

String	String Buffer
1)String is immutable	1)String Buffer is mutable
2)It is thread safe object	2)It is thread safe and synchronized
3)hashcode and equals methods are overridden	3)both methods are not availiable
4)In case of performance String is slow compare to String Buffer →String is slow and consumes more memory when we concat many strings every time it creates new instance	4)StringBuffer is fast when compare to String. →It is fast and takes less memory when u concat strings.
5)It is implementing comparable and comparator interfaces	5)It is not implementing both interfaces
6)we cannot perform these operations in String like append, insert, delete operations	6)we can perform these operations in StringBuffer like append, insert, delete operations

10. What is the difference between + operator and concat() function ?

- + operator ,one of the operands must be of the type String, else normal plus operation is executed.
- For concat(),both operands need to be strings

11. How to prove equals method is not override in StringBuffer ?

```
StringBuffer s3=new StringBuffer("hello");
StringBuffer s4=new StringBuffer("hello");
System.out.println(s3.equals(s4));//false
```

12. Difference between length and capacity of StringBuffer ?

- Length is number of characters available and capacity is total much allocated space if u add characters it will add capacity & length.

```
StringBuffer s1=new StringBuffer("hello");
int p=s1.length();
int q=s1.capacity();
System.out.println("Hello="+p);//5
System.out.println("Capacity of string Hello="+q);//16+5=21
```

13. Difference between StringBuffer and StringBuilder ?

- StringBuffer is synchronized where as StringBuilder is not synchronized.
- String builder (Multithread environment)performance is high when compare to StringBuffer(single thread environment) because no overhead of synchronization.

14. Can we use String in Switch case?

- In java 1.7,its possible to use string in switch case.

15. How do you check if two strings are equal in Java ?

- == is binary operator in java
    - It is public method of java.lang package
- == can be used for primitive types and derived types
    - Equals used for derived types.
- == you can't override this operator
    - Equals u can override in any class.
- == it will check value and location in memory
    - Equal it will check value only

16. How many strings created for String str=new String("cat") ?

- Either 1 or Either 2 because it will check cat is available in String pool. If found it will create String else it will create one string in String pool and Other string in heap memory.

17. What does String intern() method do ?

- When intern() method is invoked, if pool already contains a string equal to String object as determined by the equals(Object) method, then String from the pool is returned. Otherwise object is added to the pool and a reference to this string object is returned.

18. How to convert String to char and String to byte?

- String str="java ";
  char[] ch = str.toCharArray();
  byte[] by = str.getBytes();

19. How to create immutable class in Java ?

- Declared class as final
- Declared all fields as final
- Provide getter methods
- Provide Parameterized constructor
- Don't provide setter methods.

```java
public final class ImmutableClass {
    private final Integer id;
    public ImmutableClass(Integer id) {
        // TODO Auto-generated constructor stub
        this.id=id;
    }
    public Integer getId()
    {
        return id;
    }
    public static void main(String[] args) {
        ImmutableClass i=new ImmutableClass(10);
        System.out.println(i.getId());
    }
}
```

Regular Expressions

1. How to check string contains numeric value using Regular Expression ?
   - ".*\\d.*"
   - "[a-zA-Z]*\\d+.*"
2. How to check String contains only numeric value using Regular Expression ?
   - "\\d+"
   - "[0-9]+"
3. How to check string contains atleast one alphabets in Java ?
   - ".*[a-zA-Z]+.*"
4. How to check string contains only alphabets in java ?
   - "[a-zA-Z]+"
5. How to check string contains only special characters in Java ?
   - "[" + "-/@#!*$%^&.'_+={}()"+ "]+" ;
6. How to check string is alphanumeric or not using regular expression ?
   - "[a-zA-Z0-9]*"
7. How to validate Ip address validation using Regular Expression ?
   - private static final String IPADDRESS_PATTERN =
     "^([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\." +
     "([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\." +
     "([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\." +
     "([01]?\\d\\d?|2[0-4]\\d|25[0-5])$";

Exception Interview Questions

1. **What is Exception Handling ?**
    - Exception is an event which occurs during program execution time that stops the normal flow of execution.
    - Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, SQL Exception etc.
    - Exception are occurred due to two reasons.
        - Developer Mistakes
        - End User mistakes.
            1. While providing inputs to the application
            2. Whenever user is entered invalid data then exception occur
            3. Exception is occurred when network has disconnected at the middle of the communication
    - The main advantage of exception handling is to maintain the normal flow of the application.

2. **Difference between checked exceptions and Unchecked Exception ?**

Checked Exception	**Unchecked Exception**
1)These exceptions are checked by the compiler	1)These exceptions are not checked by the compiler
2)we can handle these exceptions by using try-catch and throws	2)These exceptions are not handled by the programmer.
3)Exception is the super class of all checked Exceptions	3)Runtime Exception is the super class of all unchecked Exceptions.

3. **Difference between Exception and Error ?**
    - Exceptions are caused due to developer mistakes these are recoverable. But errors caused due to lack of system resources these are non-recoverable.
    - StackOverflowError, OutofMemoryError, hardwareFailure

4. **Difference between throw and throws in Exception ?**

Throw	**Throws**
1) throw keyword is used to explicitly throw an exception	1) throws keyword is used to declare an exception
2) you cannot throw multiple exceptions	2) You can declare multiple exceptions.
3) throw is used within the method	3) throws is used with the method signature
4) checked exceptions cannot be propagated	4) checked exceptions can be propagated
5) throw is followed by an instance	5) throws is followed by class

5. **Explain about finally block ?**
    - Finally block is executed irrespective of try and catch.
    - It is used to provide clean-up code.
        - Database connection closing
        - Streams closing
        - Object destruction
    - Finally block will run after catch block or try block.
    - In two cases **finally block won't be executed**
        - Whenever we are giving chance to try block then only finally block will be executed otherwise it is not executed.
        - Whenever we are using system. Exit (0) JVM will be shutdown hence rest of the code won't be executed.
        - An infinite loop would also prevent a finally being called.
6. **What is exception Propagation ?**
    - Exception propagation is a way of propagating exception from method to method.
7. **Can finally block be used without catch?**
    - Yes, by try block. Finally must be followed by either try or catch
8. **Is it necessary that each try block must be followed by a catch block ?**
    - Not necessary. It should be followed by either a catch block or finally block
9. **What is the base class of Error and Exception ?**
    - Throwable
10. **What are the important methods available in Exception Class?**
    - public String getMessage()
    - public String getLocalizedMessage()
    - public synchronized Thorwable getCause()
    - public String toString()
    - public void printStackTrace()
11. **What is java7 ARM and catch block improvements ?**
    - ARM(Automatic Resource Management):
        - Most of them we use finally block just to close the resource and sometimes we forgot to close the them and get runtime Exceptions .
        - Instead of using finally block, In java 7 one feature is available ARM
        - One of the improvement is try with resources where we can create Resource in the try statement itself and use it inside the try catch block.
        - Once Execution completed, runtime environment automatically close these resources.
        - AutoCloseable interface is introduced in java7 to close the resources.
    - Benefits:
        - More Readable code and easy to write
        - Number of lines of code is reduced.
        - No need of finally block just to close these resources.

    - We can open multiple resources in try-with-Resource statement separated by a semicolon.
- We can define multiple exceptions in a single catch block in Java7

12. Difference between NoClassDefFoundError and ClassNotFoundException ?

- NoClassDefFoundError:
    - Java Test
        - If test.class file is not available then we will get runtime exception saying NoclassDefFoundError:Test
        - It available in compile time but missed in runtime.
- ClassNotFoundException:
    - We will get error when we use Class.Forname(“”)
        - Ex: Connection database to oracle but if we miss oracle jar file in Classpath it will throw ClassNotFoundException.

13. What is OutofMemoryError in java ?

- It is a subclass of java.lang.VirtualmachineError and its thrown by JVM when it ran out of heap memory.
- We can fix this error by providing more memory to run the java application using java options.

14. What are the different scenarios causing “Exception in thread main “ ?

- UnsupportedClassVersionError—JDK issue
- NoClassDefFoundError –class is not available
- NoSuchMethodError main method not available
- Predefined Exceptions-user mistake

15. What happens when exception is thrown by main method ?

- When exception is thrown by main() method, Java Runtime terminates the program and print the execution message and stack trace in system console

16. Can we have an empty catch block ?

- We can have empty catch block but it’s the poor programming.

17. What is difference between final ,finally and finalize ?

- Final and finally are the keywords and finalize is a method.
- Final is used to declare a variable ,method and class
- Finally is a block it is associated with try or catch block to close the resources
- Finalize() method is executed by garbage collector before the object is destroyed. Its a great way to make sure all the global resources are closed.