1. What is Java?

Ans: Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are many applications and websites that don't run without JAVA being installed. Java is a simple and fast, secure and reliable language.

2. What is a package in Java? List down various advantages of packages.

Ans: A java package is a group of similar types of classes , interfaces and sub-packages. Package in Java can be categorised in two form, built-in-package and user defined packages. There are many built-in packages such as java, lang, awt, javax , swing,net,io,util.

3. Explain JDK, JRE and JVM?

Ans: JDK: JDK is java development kit which is used to develop java programs and applets. It physically exists and it contains JRE+development tools.

JRE: JRE is java runtime environment which provides runtime environment using JVM. It contains set of libraries and other files that JVM uses at runtime. It physically exists.

JVM:JVM is java virtual machine which converts the source code to bytecode. It also compiles the sourcecode to bytecode.

4. Explain public static void main (String args[]) in Java.

Ans: public= It is a access modifier which gives permissions to concerned method.

Static= Static is a keyword that identifies the class related thing. It means that we don't have to create an object to access this method.

Void= It is the type of the return type. It defines what the method can return.

Main= The main is the name of the method. This method name is searched by the JVM as a starting point for an application with a particular signature only.

String args[]/String …args: it is the parameter of the main method. The argument name can be anything.

5. What are the differences between C++ and Java?

C++	JAVA
1)C++ is platform dependent.	1)JAVA is platform independent.
2)It is used for system programming.	2)It is used for application programming.
3)It is designed as systema and applications programming.	3)It is designed as an interpreter but later extended as a support network computing.
4)It supports the goto statement.	4)It doesn't support goto statement.
5)It supports multiple inheritance.	5)It doesn't support multiple inheritance.
6)It supports operator overloading.	6)It doesn't support operator overloading.
7)It uses compiler only.	7)It uses both compiler and interpreter.

6. Why Java is platform independent?
Ans: Platform independent means that once bytecode is generated, it should run on any platform(OS). So Java is platform independent because the java compiler converts the sourcecode to bytecode, which is intermediate language. Thus the bytecode can be executed on any platform(OS) using JVM.

7. What are wrapper classes in Java?
Ans: The wrapper class in java provides the mechanism to convert the primitive into object and object into primitives automatically. Since J2SE 5.0, autoboxing and unboxing feature convert primitives into objects and objects into primitives automatically. The vice-versa of autoboxing is unboxing.

8. Why pointers are not used in Java?
Ans: Java doesn't have pointer because a pointer is a variable that stores a memory address, for the purpose of acting as an alias. So java has references that refers to something else and can be used as an alias for that something else. Adding pointers also undermines security and robustness and adds the complexity of the program in java.

9. List some features of Java?
Ans: Some features of Java are:
-Simple
-Robust
-Secure

-Platform independent
-Object-oriented
-Distributed
-High performance
-Multi-threaded and interactive
-Dynamic and Extensible
-Portable
-Architectural Neutral

10. Why is Java Architectural Neutral?
Ans: This buzzword means that the program written in one program or OS is independent of other platform or environment and can run on any other operating system without recompiling them.
Byte-code is not dependent on any machine architecture and JVM can easily translate byte-code into a machine-specific code.

11. How Java enabled High Performance?
Ans: Java uses Just-in-time (JIT) compiler to compile the bytecode. This JIT compiler contains instructions that must be interpreted into instructions that can be sent directly to the processor.

12. Why Java is considered dynamic?
Ans: JAVA is considered to be dynamic because of the bytecode. The source code which is written in one platform That code can be executed in any platform. It loads the class file during the runtime only. Hence, anything that happens in runtime is dynamic.

13. What is Java Virtual Machine and how it is considered in context of Java's platform independent feature?
Ans: Java Virtual Machine(JVM) is corelated with the feature of JAVA's platform independent feature. As different computers with different operating system have their JVM's, when we submit a .class file to any operating system, JVM interpretes the bytecode into machine level language.

14. List two Java IDE's?
Ans: 1) Eclipse 2)INTELLIj
15. Why Java is called as "Platform"?
Ans: Platform is a software and hardware programs that runs. JAVA is platform independent because it is having its own JVM so that it can run on any platform. Java is platform independent.

16. Is Java Pure-Object oriented Language?

Ans: No , Java is not purely object oriented language because it supports primitive datatype such as int, byte,long,etc which are not objected oriented and , of course is what opposite of OOP is.

17. Which version of java have u learned? Name some of the new features added to it.

Ans: I have learned JAVA 8. The new features added to it are:
-permanent generation
-Parallel array sorting
-Base64 encoding and decoding
-Date & time API
-Functional Interfaces
-Lambda expressions

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

Ans: JVM gives this accessibility to JAVA for its ‘write once and run anywhere’ nature. In JAVA, The program is not converted to code directly for the hardware but it is converted into bytecode(.class), which is then interpreted bt JVM, so once compiled it generates bytecode file, which can be run anywhere. Hence it gets the WORA nature.

19. Difference between path and classpath.

Ans:

Path	Classpath
Path variable is used to set the path for all java software tools like javac.exe, java.exe, Javadoc.exe and so on.	Classpath variable is used to set the path for java classes.

20. What is the signature of main function in java ?

Ans: public static void main(String args[])

21. What is the difference between JDK and JRE?

Ans:

JDK	JRE
1)JDK is used to develop java applications which contains numerous development tools like compilers and debuggers etc.	1)JRE is the implementation of JVM and it is specially designed to execute java programs.

2) It is mainly used for executions of code and its main functionality is development.	2)It is mainly used for creating an environment for code execution.
3)It is platform-dependent.	3)It is also platform dependent.

22. What is JVM ? What it does?

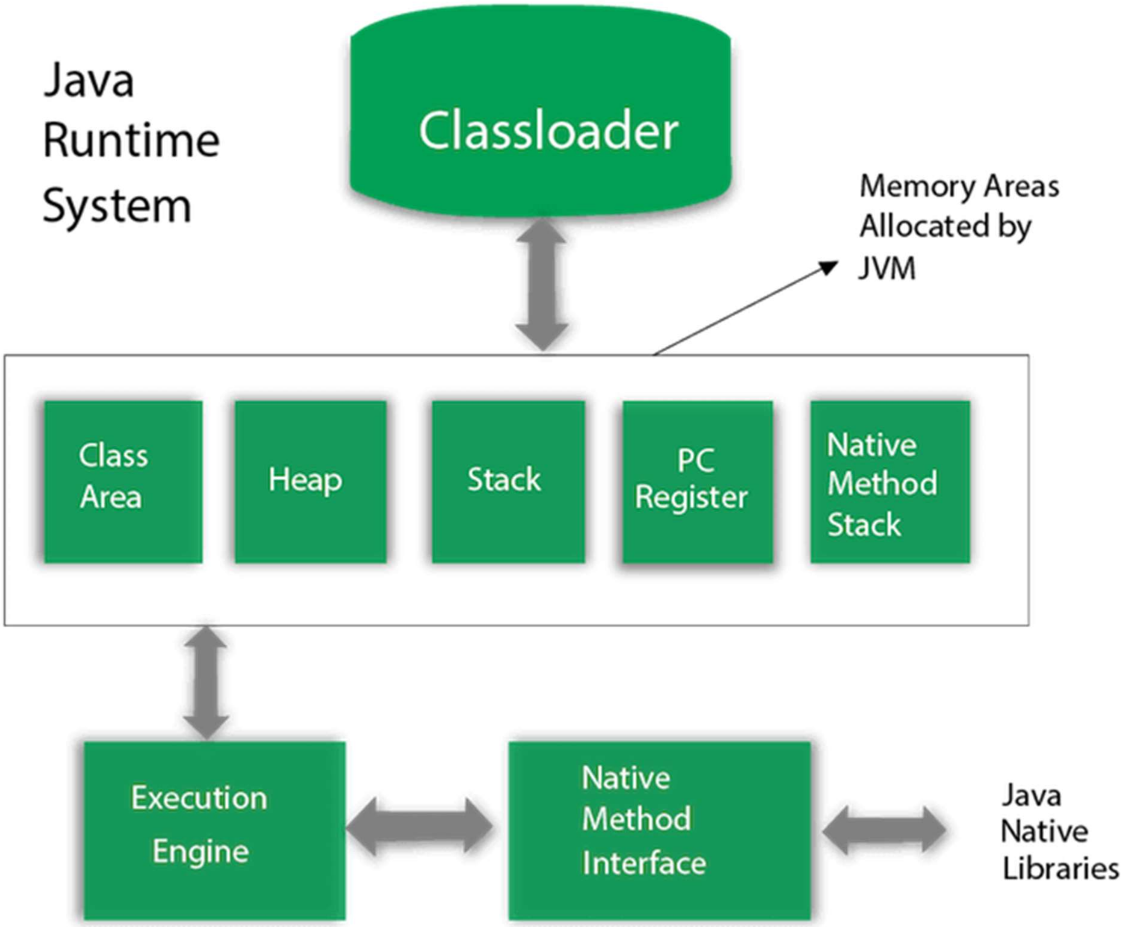
Ans: JVM means Java Virtual Machine that drives the Java Code. The main function of JVM is to convert the Java Bytecode into machine language. The Java Code is compiled to bytecode. This bytecode gets interpreted on different machines.

23. Why JVM is called as "virtual machine"?

Ans: JVM is called as "virtual machine" because it is generally implemented in software on top of a "real" hardware platform and operating system.

24. What are the main components of JVM? Explain them. Or Explain JVM Architecture.

Ans:



Class loader: It is a subsystem used to load class files. Class loader first loads the JAVA code whenever we run it.

Class Method area: In the memory, there is an area where the class data is stored during the code's execution. Class method area holds the information of static variables, static methods, static blocks, and instance methods.

Heap: The heap area is a part of the JVM memory and is created when the JVM starts up. Its size cannot be static because it increases or decreases during the application runs.

Stack: It is also referred to as thread stack. It is created for a single execution thread. The thread uses this area to store the elements like the partial result, local variable, data used for calling method and returns etc.

Native Stack: It contains the information of all the native methods used in our application.

Execution Engine: It is the central part of the JVM. Its main task is to execute the byte code and execute the Java classes. The execution engine has three main components used for executing Java classes.

- **Interpreter:** It converts the byte code into native code and executes. It sequentially executes the code. The interpreter interprets continuously and even the same method multiple times. This reduces the performance of the system, and to solve this, the JIT compiler is introduced.
- **JIT Compiler:** JIT compiler is introduced to remove the drawback of the interpreter. It increases the speed of execution and improves performance.
- **Garbage Collector:** The garbage collector is used to manage the memory, and it is a program written in Java. It works in two phases, i.e., **Mark** and **Sweep**. Mark is an area where the garbage collector identifies the used and unused chunks of memory. The Sweep removes the identified object from the **Mark**

25. What is the difference between Java compiler (javac) and JIT ?

Ans:

Java Compiler	Just in Time Compiler
It is generic	It is platform-dependent.
It compiles the whole bytecode to machine code.	It compiles the necessary portion of the bytecode to machine.

26. Is Empty .java file name a valid source file name?
Ans: Yes.It is a valid source file.

27. Is JRE different for different Platforms ?
Ans: Yes, JRE is different for different platforms because it is platform dependent.

28. Difference between C++ and java in terms of object creation.
Ans:

C++	JAVA
C++ is a compiled language.	JAVA is a compiled as well as an interpreted language.
C++ is compiled into an object code which can then be executed to produce an output.	The compiled output of a Java source code is a byte code which is platform-independent.

29. Who invokes main() function ?
Ans: When the Java interpreter executes an application (by being invoked upon the application's controlling class), it starts by calling the class's main method. The main method then calls all the other methods required to run your application.

30. What is .class file known as ?
Ans: A Java class file is a file (with the . class filename extension) containing Java bytecode that can be executed on the Java Virtual Machine (JVM).

31. Can we define more than one public class in a java source code ? what is the rule of public class and file name . ?
Ans: No, we cannot define more than one public class. The rule is that only one public class can be defined. In JAVA, class name must always be the same as file name, but sometimes file contains multiple classes.

32. What is JIT compiler?
Ans: The JIT compiler helps improve the performance of Java programs by compiling bytecodes into native machine code at run time. The JIT compiler is enabled by default. When a method has been compiled, the JVM calls the compiled code of that method directly instead of interpreting it.

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

Ans: Class(Method) Area, Heap, Stack, Program Counter Register
Native Method Stack.

34. What is the rule for local member in java.

1. Local variables cannot use any of the access level since their scope is only inside the method.
2. Final is the Only Non Access Modifier that can be applied to a local variable.
3. Local variables are not assigned a default value, hence they need to be initialized.

35. What are the various access specifiers in Java?

Ans: There are four types of access specifiers:

1)Public	Visible to all classes
2)Private	It is accessible only for that specific class.
3)Protected	It is only accessible by it’s parent class.
4)Default	It will not access outside the package.

36. What is the rule for local member in java.

1. Ans: Local variables cannot use any of the access level since their scope is only inside the method.
2. Final is the Only Non Access Modifier that can be applied to a local variable.
3. Local variables are not assigned a default value, hence they need to be initialized.

37. What is native code?

Ans: Native code refers to programming code that is configured to run on a specific processor. Native code will generally not function if used on a processor other than the one it was specifically written for unless it is allowed to run over an emulator.

38. Why there is no sizeof operator in java ?

Ans: Because the size of primitive types is explicitly mandated by the Java language. There is no variance between JVM implementations. Moreover, since allocation is done by the new operator depending on its argument there is no need to specify the amount of memory needed.

39. What kinds of programs u can develop using Java ?

- Desktop GUI Applications of Java. ...
- Mobile Applications of Java. ...
- Enterprise Applications of Java. ...
- Scientific Applications of Java. ...
- Web-based Applications of Java. ...
- Embedded Systems. ...
- Big Data Technologies. ...
- Distributed Applications of Java.

40. U have reference type as a member of class. What is the default value it gets?

Ans: The default value of reference type of member of class is null.

41. What is the job done by classloader?

Ans: Class loader has the responsibility of loading the class file into the system.

42. Explain the hierarchy of classloaders in java.

Ans: The ClassLoader Delegation Hierarchy Model always functions in the order Application ClassLoader->Extension ClassLoader->Bootstrap ClassLoader. The Bootstrap ClassLoader is always given the higher priority, next is Extension ClassLoader and then Application ClassLoader.

43. What is the role played by Bytecode Verifier ?

Ans: The bytecode verifier acts as a sort of gatekeeper: it ensures that code passed to the Java interpreter is in a fit state to be executed and can run without fear of breaking the Java interpreter. ... The types of the parameters of all bytecode instructions are known to always be correct.

44. What are the memory areas allocated by JVM ?

45. What kinds of programs u can develop using Java

46. When parseInt() method can be used?

Ans: This method is used to get the primitive data type of a certain String. parseXxx() is a static method and can have one argument or two.

47. What is finalized() method ?

Ans: Finalize() is the method of Object class. This method is called just before an object is garbage collected. finalize() method overrides to dispose system resources, perform clean-up activities and minimize memory leaks.

48. Difference between C++ pointer and Java reference.

Ans:

C++	JAVA Reference
A pointer is a variable that stores a memory address, for the purpose of acting as an alias to what is stored at that address. So, a pointer is a reference, but a reference is not necessarily a pointer.	A reference is a variable that refers to something else and can be used as an alias for that something else.

49. U have reference type as a member of class. What is the default value it gets?

50. What are the expressions allowed in switch block of java ?

Ans: The Java switch expression must be of byte, short, int, long (with its Wrapper type), enums and string. Each case statement can have a break statement which is optional.