1. What jdk contains or composed of and define each component?

In simple terms, we can say that JDK is equal to JRE and Development Tools. Components of Java Development Kit (JDK):

* **Java Archive (jar):** It is a file format based on the popular ZIP file format and is used for aggregating many files into one. Although JAR can be used as a general archiving tool, the primary motivation for its development was so that Java applets and their requisite components (.class files, images and sounds) can be downloaded to a browser in a single HTTP transaction, rather than opening a new connection for each piece. This greatly improves the speed with which an applet can be loaded onto a web page and begin functioning. The JAR format also supports compression, which reduces the size of the file and improves download time still further. Additionally, individual entries in a JAR file may be digitally signed by the applet author to authenticate their origin.
* **Java Compiler (javac):** The javac tool reads class and interface definitions, written in the Java programming language, and compiles them into bytecode class files. It can also process annotations in Java source files and classeJava Disassembler (Javap)
* **Java Debugger (jdb) :** is a simple command-line debugger for Java classes. It is a demonstration of the Java Platform Debugger Architecture that provides inspection and debugging of a local or remote Java Virtual Machine
* **Java HeaderFile Generator (javah):** javah produces C header files and C source files from a Java class. These files provide the connective glue that allows your Java and C code to interact.
* **Documentation (javadoc) :** Javadoc is a tool which comes with JDK and it is used for generating Java code documentation in HTML format from Java source code, which requires documentation in a predefined format.

1. What is IDE?

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and a debugger.

1. What are the IDEs available in the market to support java?

* Eclipse
* IntelliJ
* NetBeans
* Dr Java
* BlueJ

1. Explain the entire java life cycle.

A thread can be in one of the five states. According to sun, there is only 4 states in thread life cycle in java new, runnable, non-runnable and terminated. There is no running state.

But for better understanding the threads, we are explaining it in the 5 states.

The life cycle of the thread in java is controlled by JVM. The java thread states are as follows:

* New
* Runnable
* Running
* Non-Runnable (Blocked)
* Terminated

1. What is class?

A class is the blueprint from which individual objects are created.

1. What is object?

Object is an instance of a class. It has states and behaviors.

1. What is the entry point of an application?

The entry point is the main function.

1. Why main is static?

This is necessary because main method is called by the JVM before any objects are made. Since it is static, it can be directly invoked via the class without any objects.

1. What is the difference between an Interface and Abstract class?

* Abstract classes can have constants, members, method stubs (methods without a body) and defined methods, whereas interfaces can only have constants and methods stubs.
* Methods and members of an abstract class can be defined with any visibility, whereas all methods of an interface must be defined as public (they are defined public by default).
* When inheriting an abstract class, a concrete child class must define the abstract methods, whereas an abstract class can extend another abstract class and abstract methods from the parent class don't have to be defined.
* Similarly, an interface extending another interface is not responsible for implementing methods from the parent interface. This is because interfaces cannot define any implementation.
* A child class can only extend a single class (abstract or concrete), whereas an interface can extend or a class can implement multiple other interfaces.
* A child class can define abstract methods with the same or less restrictive visibility, whereas a class implementing an interface must define the methods with the exact same visibility (public).

1. Can you make an Object from an Interface and Abstract class? if not how do you use it ?

You cannot create an object in an interface nor abstract class. This is a logical limitation because since interface has undefined methods and abstract class may also have undefined methods, and if an object gets instantiated and then call on an unimplemented method, you will receive an error message.

1. What is Access Specifier?

Access specifiers are used to control the visibility of members like classes, variables and methods. There are three access specifiers: public, private and protected. It is used to facilitate the encapsulation of components.

1. What is OOP ? Define each feature of java OOP.

In Java, a class is a definition of objects of the same kind. In other words, a class is a blueprint, template, or prototype that defines and describes the static attributes and dynamic behaviors common to all objects of the same kind.

An instance is a realization of a particular item of a class. In other words, an instance is an instantiation of a class. All the instances of a class have similar properties, as described in the class definition. For example, you can define a class called "Student" and create three instances of the class "Student" for "Peter", "Paul" and "Pauline".

The term "object" usually refers to instance. But it is often used loosely, and may refer to a class or an instance.

1. What is API? Name list of API that you have used so far.

An application programming interface (API) is a particular set of rules ('code') and specifications that software programs can follow to communicate with each other. It serves as an interface between different software programs and facilitates their interaction.

Used api: Google, Facebook graph

1. Does java support multiple inheritance?

It does not support multiple inheritance because two classes may define different ways of doing the same thing, and the subclass can't choose which one to pick.

1. What is method overloading and when it happens?

Method overriding allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its superclasses or parent classes. The implementation in the subclass overrides (replaces) the implementation in the superclass by providing a method that has same name, same parameters or signature, and same return type as the method in the parent class.

1. What is method overriding and when it happens?

Method overloading is the ability to create multiple methods of the same name with different implementations. Calls to an overloaded function will run a specific implementation of that function appropriate to the context of the call, allowing one function call to perform different tasks depending on context.

For example, doTask() and doTask(object O) are overloaded methods. To call the latter, an object must be passed as a parameter, whereas the former does not require a parameter, and is called with an empty parameter field.

1. Explain exceptions in java and how to handle it.

Exceptions are events that occur during the execution of programs that disrupt the normal flow of instructions (e.g. divide by zero, array access out of bound, etc.).

Exception Handling:

* Try-catch
* Nested Try Catch.
* Checked and unchecked exceptions.
* Finally block
* Try-catch-finally.
* Finally block & return statement.
* Throw exception

1. What is static keyword in java? How it has been used in variables and methods?

A static member is a member of a class that isn’t associated with an instance of a class. Instead, the member belongs to the class itself. As a result, you can access the static member without first creating a class instance.

* The value of a static variable is the same across all instances of the class. In other words, if a class has a static variable named CompanyName, all objects created from the class will have the same value for CompanyName.
* Static methods are associated with the class itself, not with any particular object created from the class. As a result, you don’t have to create an object from a class before you can use static methods defined by the class.

1. What is final and how it has been used variables and methods?

The final keyword is use to make sure that values assigned to variables cannot be changed. If a method is declared final, that means the method cannot be changed/overridden.

1. What is final, finally and finalize?

* "Final" denotes that something cannot be changed. Final can be used to mark a variable "unchangeable” and also make a method not "overrideable". Final can also make a class not "inheritable". i.e. the class cannot be subclassed.
* "Finally" is used in conjunction with a try/catch block. Anything inside of the "finally" clause will be executed regardless of if the code in the 'try' block throws an exception or not.
* "Finalize" is called by the JVM before an object is about to be garbage collected.

1. What is a constructor?

A constructor in Java is a block of code similar to a method that's called when an instance of an object is created. Here are the key differences between a constructor and a method: A constructor doesn't have a return type. The name of the constructor must be the same as the name of the class.

1. What is "this" keyword in java?

“This” is a keyword in Java. It can be used inside the Method or constructor of Class. It works as a reference to the current Object whose Method or constructor is being invoked. This keyword can be used to refer to any member of the current object from within an instance Method or a constructor.

1. What is "super" keyword in java?

“Super” is a keyword. It is used inside a sub-class method definition to call a method defined in the super class. Private methods of the super-class cannot be called. Only public and protected methods can be called by the super keyword. It is also used by class constructors to invoke constructors of its parent class.

1. What is JVM stand for?

Java virtual machine

1. What version of java are you using?

java version "1.8.0\_121"

1. What is JAR stand for?

Java Archive

1. What is compile time and run time?

The source code must be compiled into machine code in order to become and executable program. This compilation process is referred to as compile time.

A compiled program can then be opened and run by a user. When an application is running, it is called runtime.

1. What is heap?

The heap is the runtime data area from which memory for all class instances and arrays is allocated. The heap is created on virtual machine start-up.

1. How java manage its memory?

* Java objects reside in an area called the heap. The heap is created when the JVM starts up and may increase or decrease in size while the application runs. When the heap becomes full, garbage is collected. During the garbage collection process, objects that are no longer used are cleared, thus making space for new objects.
* During object allocation, the JVM distinguishes between small and large objects. The limit for when an object is considered large depends on the JVM version, the heap size, the garbage collection strategy and the platform used
* Permanent Generation or “Perm Gen” contains the application metadata required by the JVM to describe the classes and methods used in the application.
* Memory Pools are created by JVM memory managers to create a pool of immutable objects, if implementation supports it. String Pool is a good example of this kind of memory pool. Memory Pool can belong to Heap or Perm Gen, depending on the JVM memory manager implementation.

1. What is the difference between String, StringBuffer and StringBuilder?

* Strings are a sequence of characters. In the Java programming language, strings are objects.
* StringBuffer methods are synchronized (thread safe) while StringBuilder are not.
* StringBuilder is faster than StringBuffer because it's not synchronized.
* The operations are "almost" the same, but using synchronized methods in a single thread is overkill (this would be inefficient use of resources.).