Question: What is Java?

Answer:
Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to data-centers, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

Notes

Question: Mention some features of Java?

Answer: Some of the features which play important role in the popularity of java are as follows:

1.Object-Oriented: Java is a object oriented programming language. Everything in Java is an Object.
2.Portable: Java run time environment uses a bytecode verification process to make sure that code loaded over the network doesn’t violate Java security constraints.
3.Platform independent: Java is platform independent. Java is a write once, run anywhere language. Without any modifications, we can use a program in different platforms.
4.Secured: Java is well known for its security. It delivers virus free systems.
5.High Performance: Java enables high performance with the use of JIT (Just-In-Time) compilers
6.Multithreaded: Java Multithreaded features allows us to write programs that can perform many tasks simultaneously. Multithreading concept of Java shares a common memory area. It doesn’t occupy memory for each thread.

Notes

Question: What is the difference between Declaration and Definition in Java?

Answer:

1.Declaration: If you just declare a class or method/function or variable without mentioning anything about what that class or method/ function or variable looks like is called as declaration in Java.

2.Definition: If you define how a class or method/function or variable is implemented then it is called definition in Java.

3.When we create an interface or abstract class, we simply declare a method/function but not define it.

Notes

Question: What is an Object in Java?

Answer:
An object is an instance of a class. Objects have state (variables) and behavior (methods).
Example: A dog is an object of Animal class. The dog has its states such as color, name, breed, and behaviors such as barking, eating, wagging her tail.

Notes

Question: What is a Class in Java?

Answer:
A class can be defined as a collection of objects. It is the blueprint or template that describes the state and behavior of an object.

Notes

Question: What is Constructor in Java?

Answer:

Constructor in Java is used in the creation of an Object that is an instance of a Class. Constructor name should be same as class name. It looks like a method but its not a method. It wont return any value. We have seen that methods may return a value. If there is no constructor in a class, then compiler automatically creates a default constructor.

Notes

Question: What is Local Variable and Instance Variable?

Answer:

Local Variable: Local variable is a variable which we declare inside a Method. A method will often store its temporary state in local variables.

Instance Variable (Non-static): Instance variable is a variable which is declared inside a Class but outside a Method. We don’t declare this variable as Static because these variables are non-static variables.

Notes

Question: What are the OOPs concepts?

Answer:

OOPS Stands for Object Oriented Programming System. It includes Abstraction, Encapsulation, Inheritance, Polymorphism, Interface etc.,

Notes

Question: What is Inheritance in Java?

Answer:
Inheritance is a process where one class inherits the properties of another class.

Notes

Question: What is Polymorphism?

Answer:
Polymorphism allows us to perform a task in multiple ways. Let’s break the word Polymorphism and see it, ‘Poly’ means ‘Many’ and ‘Morphos’ means ‘Shapes’.
Assume we have four students and we asked them to draw a shape. All the four may draw different shapes like Circle, Triangle, and Rectangle.

Notes

Question: What are the types of Polymorphism?

Answer:

There are two types of Polymorphism in Java

1. Compile time polymorphism (Static binding) – Method overloading
2. Runtime polymorphism (Dynamic binding) – Method overriding

We can perform polymorphism by ‘Method Overloading’ and ‘Method Overriding’

A class having multiple methods with same name but different parameters is called Method Overloading

There are three ways to overload a method.

1.Parameters with different data types

2.Parameters with different sequence of a data types

3.Different number of parameters

Notes

Question: What is Method Overriding?

Answer:
Declaring a method in child class which is already present in the parent class is called Method Overriding.
In simple words, overriding means to override the functionality of an existing method.
In this case, if we call the method with child class object, then the child class method is called. To call the parent class method we have to use **super** keyword.

Notes

Question: What is Abstraction in Java?

Answer:
Abstraction is the methodology of hiding the implementation of internal details and showing the functionality to the users.
Example: Mobile Phone.
A layman who is using mobile phone doesn’t know how it works internally but he can make phone calls.

Notes

Question: What is Abstract Class in Java?

Answer:

We can easily identify whether a class is an abstract class or not. A class which contains abstract keyword in its declaration then it is an Abstract Class.

Points to remember:

1. Abstract classes may or may not include abstract methods
2. If a class is declared abstract then it cannot be instantiated.
3. If a class has abstract method then we have to declare the class as abstract class
4. When an abstract class is subclassed, the subclass usually provides implementations for all of the abstract methods in its parent class. However, if it does not, then the subclass must also be declared abstract.

Notes

Question: What is Abstract Method?

Answer:
An abstract method is a method that is declared without an implementation (without braces, and followed by a semicolon), like this:
In order to use an abstract method, you need to override that method in sub class.

Notes

Question: What is Interface in Java?

Answer:
An interface in Java looks similar to a class but both the interface and class are two different concepts. An interface can have methods and variables just like the class but the methods declared in interface are by default abstract. We can achieve 100% abstraction and multiple inheritance in Java with Interface. Read more on

Notes

Question: What is Encapsulation in Java?

Answer:
Encapsulation is a mechanism of binding code and data together in a single unit. Let’s take an example of Capsule. Different powdered or liquid medicines are encapsulated inside a capsule. Likewise in encapsulation, all the methods and variables are wrapped together in a single class. Read more on

Notes

Question: Difference between Array and ArrayList?

Answer:

ARRAY	ARRAYLIST
Array is static	ArrayList is dynamic
Size of the array should be given at the time of array declaration. We cannot change the size of array after creating it	Size of the array may not be required. It changes the size dynamically. Capacity of ArrayList increases automatically whenever we add elements to an ArrayList
Array can contain both primitive data types as well as objects	ArrayList cannot contain primitive data types. It contains only objects
Arrays are multidimensional	ArrayList is always single dimension

Notes

Question: Difference between ArrayList and HashSet in Java?

Answer:

ARRAYLIST	HASHSET
ArrayList implements List interface	HashSet implements Set interface
ArrayList allows duplicates	HashSet doesn't allow duplicates
ArrayList is an ordered collection and maintains insertion order of elements	HashSet is an unordered collection and doesn't maintain insertion order
ArrayList is backed by an Array	HashSet is backed by an HashMap instance
ArrayList is an index based	HashSet is object based
In ArrayList, we can retrieve object by calling get() method or remove object by calling remove() method	In HashSet, we can't achieve get() method

Notes

Question: What are the different access modifiers available in Java?

Answer:

Access modifiers are subdivided into four types such as Default, Public, Private, Protected

default: The scope of default access modifier is limited to the package only. If we do not mention any access modifier, then it acts like a default access modifier.

private: The scope of private access modifier is only within the classes.

Note: Class or Interface cannot be declared as private

protected: The scope of protected access modifier is within a package and also outside the package through inheritance only.

Note: Class cannot be declared as protected

public: The scope of public access modifier is everywhere. It has no restrictions. Data members, methods and classes that declared public can be accessed from anywhere.

Notes

Question: Difference between static binding and dynamic binding?

Answer:

1. Static binding is also known as early binding whereas dynamic binding is also known as late binding.
2. Determining the type of an object at compile time is Static binding whereas determining the type of an object at run time is dynamic binding
3. Java uses static binding for overloaded methods and dynamic binding for overridden methods.

Notes

Question: Difference between Abstract Class and Interface?

Answer:

ABSTRACT CLASS	INTERFACE
To declare Abstract class we have to use abstract keyword	To declare Interface we have to use interface keyword
In an Abstract class keyword abstract is mandatory to declare a method as an abstract	In an Interface keyword abstract is optional to declare a method as an abstract. Compiler treats all the methods as abstract by default
An abstract class contains both abstract methods and concrete methods (method with body)	An interface can have only abstract methods
An abstract class provides partial abstraction	An interface provides fully abstraction
An abstract class can have public and protected abstract methods	An interface can have only public abstract methods
An abstract class can have static, final or static final variables with any access modifiers	An interface can have only public static final variables
An abstract class can extend one class or one abstract class	An interface can extend any number of interfaces
Abstract class doesn't support multiple inheritance	Interface supports multiple inheritance

Notes

Question: What is Multiple Inheritance?

Answer:
If a class implements multiple interfaces, or an interface extends multiple interfaces then it is known as multiple inheritance.

Notes

Question: What are the Java IDE’s?

Answer:
Eclipse and NetBeans are the IDE's of JAVA.
An integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development.

Notes

Question: What is Exception?

Answer:

An Exception is a problem that can occur during the normal flow of an execution. A method can throw an exception when something wails at runtime. If that exception couldn’t be handled, then the execution gets terminated before it completes the task.
If we handled the exception, then the normal flow gets continued. Exceptions are a subclass of java.lang.Exception.

Notes

Question: What are the types of Exceptions?

Answer:

Two types of Exceptions are explained below in detail.

Checked Exception:

These exceptions are checked by the compiler at the time of compilation. Classes that extend Throwable class except Runtime exception and Error are called checked Exception.

Checked Exceptions must either declare the exception using throes keyword (or) surrounded by appropriate try/catch.

E.g. ClassNotFound Exception

Unchecked Exception:

These exceptions are not checked during the compile time by the compiler. The compiler doesn't force to handle these exceptions.

It includes: Arithmetic Exception, ArrayIndexOutOfBounds Exception

Notes

Question: What are the types of Exceptions?

Answer:

Two different ways to handle exception are explained below:

1) Using try/catch:

A risky code is surrounded by try block. If an exception occurs, then it is caught by the catch block which is followed by the try block.

2) By declaring throws keyword:

At the end of the method, we can declare the exception using throws keyword.

Notes

Question: What are the Advantages of Exception handling?

Answer:

Given below are the advantages:

- The normal flow of the execution won’t be terminated if exception got handled
- We can identify the problem by using catch declaration

Notes

Question: What are Exception handling keywords in Java?

Answer:
Given below are the two Exception Handling Keywords:
try:
When a risky code is surrounded by a try block. An exception occurring in the try block is caught by a catch block. Try can be followed either by catch (or) finally (or) both. But any one of the blocks is mandatory.
catch:
This is followed by try block. Exceptions are caught here.
finally:
This is followed either by try block (or) catch block. This block gets executed regardless of an exception. So generally clean up codes are provided here.

Notes

Question: What is the final keyword in Java?

Answer:

Final variable:

Once a variable is declared as final, then the value of the variable could not be changed. It is like a constant.

Example: final int = 12;

Final method:

A final keyword in a method that couldn't be overridden. If a method is marked as a final, then it can't be overridden by the subclass.

Final class:

If a class is declared as final, then the class can not be inherited from. No class can extend the final class.

Notes

Question: What is a Thread?

Answer:

In Java, the flow of an execution is called Thread. Every java program has at least one thread called main thread, the Main thread is created by JVM. The user can define their own threads by extending Thread class (or) by implementing Runnable interface. Threads are executed concurrently.

Notes

Question: How do you make a thread in Java?

Answer:

1) Extend Thread class:

Extending a Thread class and override the run method. The thread is available in java.lang.thread.

The disadvantage of using a thread class is that we cannot extend any other classes because we have already extend the thread class. We can overload the run () method in our class.

2) Implement Runnable interface:

Another way is implementing the runnable interface. For that we should provide the implementation for run () method which is defined in the interface.

Notes

Question: What does yield method of the Thread class do?

Answer:
A yield () method moves the currently running thread to a runnable state and allows the other threads for execution. So that equal priority threads have a chance to run. It is a static method. It doesn’t release any lock.
Yield () method moves the thread back to the Runnable state only, and not the thread to sleep (), wait () (or) block.

Notes

Question: Explain about wait () method.

Answer:

wait () method is used to make the thread to wait in the waiting pool. When a wait () method is executed during a thread execution then immediately the thread gives up the lock on the object and goes to the waiting pool. Wait () method tells the thread to wait for a given amount of time.

Then the thread will wake up after notify () (or) notify all () method is called.

Wait() and the other above-mentioned methods do not give the lock on the object immediately until the currently executing thread completes the synchronized code. It is mostly used in synchronization.

Notes

Question: Difference between notify() method and notifyAll() method in Java.

Answer:

Given below are few differences between notify() method and notifyAll() method

notify()	**notifyAll()**
This method is used to send a signal to wake up a single thread in the waiting pool.	This method sends the signal to wake up all the threads in a waiting spool.

Notes

Question: How to stop a thread in java? Explain about sleep () method in a thread?

Answer:

We can stop a thread by using the following thread methods.

- Sleeping
- Waiting
- Blocked

Sleep:

Sleep () method is used to sleep the currently executing thread for the given amount of time. Once the thread is wake up it can move to the runnable state. So sleep () method is used to delay the execution for some period.

It is a static method.

Example:

Thread. Sleep (2000)

So it delays the thread to sleep 2 milliseconds. Sleep () method throws an uninterrupted

Notes

Question: When to use Runnable interface Vs Thread class in Java?

Answer:
If we need our class to extend some other classes other than the thread then we can go with the runnable interface because in java we can extend only one class.
If we are not going to extend any class then we can extend the thread class.

Notes

Question: Difference between start() and run() method of thread class.

Answer:
Start() method creates new thread and the code inside the run () method is executed in the new thread. If we directly called the run() method then a new thread is not created and the currently executing thread will continue to execute the run() method.

Notes

Question: What is Multi-threading?

Answer:
Multiple threads are executed simultaneously. Each thread starts their own stack based on the flow (or) priority of the threads.
On the 1st line execution, JVM calls the main method and the main thread stack looks as shown below.

Notes

Question: What is Synchronization?

Answer:
Synchronization makes only one thread to access a block of code at a time. If multiple thread accesses the block of code, then there is a chance for inaccurate results at the end. To avoid this issue, we can provide synchronization for the sensitive block of codes.
The synchronized keyword means that a thread needs a key in order to access the synchronized code.
Locks are per objects. Every Java object has a lock. A lock has only one key. A thread can access a synchronized method only if the thread can get the key to the objects lock.
For this, we use “Synchronized” keyword.

Notes

Question: What is the disadvantage of Synchronization?

Answer:
Synchronization is not recommended to implement all the methods. Because if one thread accesses the synchronized code then the next thread should have to wait. So it makes slow performance on the other end.

Notes

Question: What is meant by Serialization?

Answer:
Converting a file into a byte stream is known as Serialization. The objects in the file is converted to the bytes for security purposes. For this, we need to implement java.io.Serializable interface. It has no method to define.
Variables that are marked as transient will not be a part of the serialization. So we can skip the serialization for the variables in the file by using a transient keyword.

Notes

Question: What is the purpose of a transient variable?

Answer:

Ans: Transient variables are not part of the serialization process. During deserialization, the transient variables values are set to default value. It is not used with static variables.

Example:

transient int numbers;

Notes

Question: Which methods are used during Serialization and Deserialization process?

Answer:

ObjectOutputStream and ObjectInputStream classes are higher level java.io. package. We will use them with lower level classes FileOutputStream and FileInputStream.
ObjectOutputStream.writeObject —->Serialize the object and write the serialized object to a file.
ObjectInputStream.readObject —> Reads the file and deserializes the object.
To be serialized, an object must implement the serializable interface. If superclass implements Serializable, then the subclass will automatically be serializable.

Notes

Question: What is the purpose of a Volatile Variable?

Answer:
Volatile variable values are always read from the main memory and not from thread's cache memory. This is used mainly during synchronization. It is applicable only for variables.

Example:
volatile int number;

Notes