**OOP Interview Questions**

difference between interfaces and abstarct classes and when to use one over another

 Main difference is methods of a Java interface are implicitly abstract and cannot have implementations. A Java abstract class can have instance methods that implements a default behavior.

 Variables declared in a Java interface is by default final. An abstract class may contain non-final variables.

 Members of a Java interface are public by default. A Java abstract class can have the usual flavors of class members like private, protected, etc.

 Java interface should be implemented using keyword “implements”; A Java abstract class should be extended using keyword “extends”.

 An interface can extend another Java interface only an abstract class can extend another Java class and implement multiple Java interfaces.

 A Java class can implement multiple interfaces but it can extend only one abstract class.

 Interface is absolutely abstract and cannot be instantiated; A Java abstract class also cannot be instantiated, but can be invoked if a main() exists.

 In comparison with java abstract classes, java interfaces are slow as it requires extra indirection.

how to stop a class from being inherited (selaed classes and private consructor)

/////////////////////C#

In order to prevent a class in C# from being inherited, the keyword sealed is used. Thus a sealed class may not serve as a base class of any other class. It is also obvious that a sealed class cannot be an abstract class. Code below...

//C# Example

sealed class ClassA

{

public int x;

public int y;

}

///////////////////////JAVA

Officially, the Java language provides the keyword 'final' that is supposed to fulfill this task. Consider the following code sample:

//FinalDemo.java

public final class FinalDemo {

}

Private Constructor is a special instance constructor present in C# language. Basically, private constructors are used in class that contains only static members. The private constructor is always declared by using a private keyword.

Important points:

* It is the implementation of a singleton class pattern.
* Use private constructor when class have only static members.
* Using private constructor, prevents the creation of the instances of that class.
* If a class contains only private constructor without parameter, then it prevents the automatic generation of default constructor.
* If a class contains only private constructors and does not contain public constructor, then other classes are not allowed to create instances of that class except nested class.

what is encapsulation explain it

It refers to the bundling of data with the methods that operate on that data. Encapsulation is used to hide the values or state of a structured data object inside a class, preventing unauthorized parties’ direct access to them. Publicly accessible methods are generally provided in the class (so-called getters and setters) to access the values, and other client classes call these methods to retrieve and modify the values within the object.

what is abstraction explain it

Abstraction is the process of taking away or removing characteristics from something in order to reduce it to a of essential characteristics. Through the process of abstraction, a programmer hides all but the relevant data about an [object](https://searchapparchitecture.techtarget.com/definition/object) in order to reduce complexity and increase efficiency.

Making coffee with a coffee machine is a good example of abstraction. You need to know how to use your coffee machine to make coffee. You need to provide water and coffee beans, switch it on and select the kind of coffee you want to get. The thing you don’t need to know is how the coffee machine is working internally to brew a fresh cup of delicious coffee. You don’t need to know the ideal temperature of the water or the amount of ground coffee you need to use.

what is polymorphism, give an example

The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form. A real-life example of polymorphism, a person at the same time can have different characteristics. Like a man at the same time is a father, a husband, an employee. So the same person possess different behavior in different situations. This is called polymorphism.

polymorphism is mainly divided into two types:

* Compile time Polymorphism
* Runtime Polymorphism

**Compile time polymorphism**: This type of polymorphism is achieved by function overloading or operator overloading.

* [Function Overloading](https://www.geeksforgeeks.org/function-overloading-c/): When there are multiple functions with same name but different parameters then these functions are said to be **overloaded**. Functions can be overloaded by **change in number of arguments** or/and **change in type of arguments**.

[Runtime polymorphism](https://www.geeksforgeeks.org/virtual-functions-and-runtime-polymorphism-in-c-set-1-introduction/): This type of polymorphism is achieved by Function Overriding.

* [Function overriding](https://www.geeksforgeeks.org/override-keyword-c/) on the other hand occurs when a derived class has a definition for one of the member functions of the base class. That base function is said to be **overridden**.

difference between static and dynamic polymorphism

Static polymorphism is polymorphism that occurs at compile time(early binding), and dynamic polymorphism is polymorphism that occurs at runtime (during application execution/ late binding).

1. Static binding/Compile-Time binding/Early binding/Method overloading.(in same class)

2. Dynamic binding/Run-Time binding/Late binding/Method overriding.(in different classes)

what is multi threading, how to achive that

Multitasking: Ability to execute more than one task at the same time is known as multitasking.

Multithreading: We already discussed about it. It is a process of executing multiple threads simultaneously. Multithreading is also known as Thread-based Multitasking.

Multiprocessing: It is same as multitasking, however in multiprocessing more than one CPUs are involved. On the other hand one CPU is involved in multitasking.

Parallel Processing: It refers to the utilization of multiple CPUs in a single computer system.

## Creating a thread in Java

There are two ways to create a thread in Java:  
1) By extending Thread class.  
2) By implementing Runnable interface.

what are stack and heaps

Stack is used for static memory allocation and Heap for dynamic memory allocation, both stored in the computer's RAM .

Key Differences Between Stack and Heap Allocations

1. In a stack, the allocation and deallocation is automatically done by whereas, in heap, it needs to be done by the programmer manually.
2. Handling of Heap frame is costlier than handling of stack frame.
3. Memory shortage problem is more likely to happen in stack whereas the main issue in heap memory is fragmentation.
4. Stack frame access is easier than the heap frame as the stack have small region of memory and is cache friendly, but in case of heap frames which are dispersed throughout the memory so it cause more cache misses.
5. Stack is not flexible, the memory size allotted cannot be changed whereas a heap is flexible, and the allotted memory can be altered.
6. Accessing time of heap takes is more than a stack.

when is a reference variable

# References in C++

Last Updated: 28-04-2020

When a variable is declared as reference, it becomes an alternative name for an existing variable. A variable can be declared as reference by putting ‘&’ in the declaration.

|  |
| --- |
| int x = 10;  // ref is a reference to x.    int& ref = x;   // Value of x is now changed to 20    ref = 20;    cout << "x = " << x << endl ;    // Value of x is now changed to 30    x = 30;    cout << "ref = " << ref << endl ; |

Output:

x = 20

ref = 30

what is pass be value and pass by reference. Example

**Pass by reference** (also called pass by address) means to pass the reference of an argument in the calling function to the corresponding formal parameter of the called function so that a copy of the address of the actual parameter is made in memory, i.e. the caller and the callee use the same variable for the parameter. If the callee modifies the parameter variable, the effect is visible to the caller’s variable.

**Pass by value** means that a copy of the actual parameter’s value is made in memory, i.e. the caller and callee have two independent variables with the same value. If the callee modifies the parameter value, the effect is not visible to the caller.

how value type and reference types go in memory -examples of value types and reference types

## Value Type

## A data type is a value type if it holds a data value within its own memory space. It means the variables of these data types directly contain values. When you pass a value-type variable from one method to another, the system creates a separate copy of a variable in another method. If value got changed in the one method, it wouldn't affect the variable in another method.

For example, consider integer variable int i = 100;

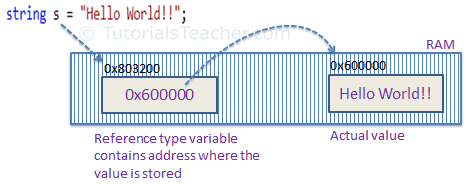
## Reference Type

## Unlike value types, a reference type doesn't store its value directly. Instead, it stores the address where the value is being stored. In other words, a reference type contains a pointer to another memory location that holds the data.

For example, consider the following string variable:

string s = "Hello World!!";

The following image shows how the system allocates the memory for the above string variable.

[](https://www.tutorialsteacher.com/Content/images/csharp/raference-type-memory-allocation.png)Memory Allocation of Reference Type Variable

As you can see in the above image, the system selects a random location in memory (0x803200) for the variable s. The value of a variable s is 0x600000, which is the memory address of the actual data value. Thus, reference type stores the address of the location where the actual value is stored instead of the value itself.

what things go in stack and what things go in heap

what is garbage collection

Java garbage collection is the process by which Java programs perform automatic memory management. Java programs compile to bytecode that can be run on a Java Virtual Machine, or JVM for short. When Java programs run on the JVM, objects are created on the heap, which is a portion of memory dedicated to the program. Eventually, some objects will no longer be needed. The garbage collector finds these unused objects and deletes them to free up memory.

what does finally do

Java finally block is a block that is used to execute important code such as closing connection, stream etc. Java finally block is always executed whether exception is handled or not. Java finally block

Question: if you create a class and define a character inside it what will be the total memory it will take and what memory will be taken for an empty class

Empty class takes about 1byte

what is immutability?

Immutability is a design pattern where something can't be modified after being instantiated. If we want to change the value of that thing, we must recreate it with the new value instead.

are string immutable? if yes how?

The string is Immutable in Java because String objects are cached in the String pool. Since cached String literals are shared between multiple clients there is always a risk, where one client's action would affect all another client. For example, if one client changes the value of the String "Test" to "TEST", all other clients will also see that value as explained in the first example.

-Can a parrent class' reference variable have a child class's object and vice versa?

-Can we make an object of interface?

We can't create instance(interface can't be instantiated) of interface but we can make reference of it that refers to the Object of its implementing class. A class can implement more than one interface. ... A class that implements interface must implements all the methods in interface

-static classes vs non static classes

Non-static classes can be instantiated, whereas static classes cannot be instantiated i.e. you cannot use the new keyword to create a variable of the class type.

Non-static classes can have instance method and static methods.

Access the members of a static class by using the class name itself, whereas Static class is sealed.

-static methods vs non static methods

A static method belongs to the class, and you do not have to create an instance of the class to access the static method. ... A non-static method belongs to an object of the class, and you have to create an instance of the class to access the non-static method.

what are inner classes?

In object-oriented programming, an inner class or nested class is a class declared entirely within the body of another class or interface. It is distinguished from a subclass.

-try creating static inner classes in non static classes and vice versa

Java static nested class example with instance method

A static class i.e. created inside a class is called static nested class in java. It cannot access non-static data members and methods. It can be accessed by outer class name.

* It can access static data members of outer class including private.
* Static nested class cannot access non-static (instance) data member or method.

1. class TestOuter1{
2. static int data=30;
3. static class Inner{
4. void msg(){System.out.println("data is "+data);}
5. }
6. public static void main(String args[]){
7. TestOuter1.Inner obj=new TestOuter1.Inner();
8. obj.msg();
9. }
10. }

Output:

data is 30

In this example, you need to create the instance of static nested class because it has instance method msg(). But you don't need to create the object of Outer class because nested class is static and static properties, methods or classes can be accessed without object.

can interfaces have variables and properties?

An interface can have methods and variables just like the class but the methods declared in interface are by default abstract (only method signature, no body, see: Java abstract method). Also, the variables declared in an interface are public, static & final by default.

-if class A is inheriting class B, which constuctor will be call first A or B?

B (Parent class) class constructor will be called first but points to remember that, the constructor of base class is :

1. default or non-parameterized constructor or
2. parameterized constructor

If it is default or non - parameterized constructor then there is nothing to do because compiler itself generate a call to base class constructor.

-List and Describe some OOP design patterns

Design pattern is a solution approach to a common problem, It should be an industry standard without language dependent Design Pattern can be classified into three types

Creational design patterns

Creational design patterns are concerned with**the way of creating objects.** These design patterns are used when a decision must be made at the time of the instantiation of a class (i.e. creating an object of a class). This pattern can be further divided into class-creation patterns and object-creational patterns. While class-creation patterns use inheritance effectively in the instantiation process, object-creation patterns use delegation effectively to get the job done.

Structural design patterns

Structural design patternsare concerned with how classes and objects can be composed, to form larger structures. The structural design patterns simplify the structure by identifying relationships. These patterns focus on, how the classes inherit from each other and how they are composed of other classes. Structural class-creation patterns use inheritance to compose interfaces. Structural object-patterns define ways to compose objects to obtain new functionality.

Behavioral design patterns

Behavioral design patterns are concerned with the interaction and responsibility of objects. In these design patterns, the interaction between the objects should be in such a way that they can easily talk to each other and still should be loosely coupled.That means the implementation and the client should be loosely coupled to avoid hard coding and dependencies.

Implement Singelton Pattern

A singleton in Java is a class for which only one instance can be created. It provides a global point of access this instance.

public class Singleton {

private static Singleton uniqInstance;

private Singleton() { }

public static Singleton getInstance()

{

if (uniqInstance == null)

{

synchronized(Singleton.class)

{ *// check again to avoid multi-thread access*

if (uniqInstance == null) uniqInstance = new Singleton();

}

}

return uniqInstance; } *// other useful methods here* }

## The observer pattern

The observer pattern defines a one-to-many dependency between objects so that when one object changes state, all of its dependents are notified and updated automatically.

The object which is being watched is called the subject. The objects which are watching the state changes are called observers or listeners.

Façade Pattern

The Facade Pattern provides a unified interface to a set of interfaces in as subsystem. Facade defines a higher-level interface that makes the subsystem easier to use.

The Facade Pattern leaves the subsystem accessible to be used directly.

### Example

Assume you have a database access class with different methods to read the different tables. The client requires the complete result. You could use a facade pattern which hides the complex database access interface behind a few easy to understand and maintainable interface, e.g., load() and get()

## the adapter pattern

The adapter pattern is widely known in software development and used in many programming languages, e.g., Java. The adapter pattern describes how to convert an object into another object which a clients expects. This pattern mainly adapts one object to another one. Adapters allow objects to work together that couldn’t otherwise because of incompatible interfaces. Adapter allows to reuse existing coding without changing it, as the adapter ensures the conversion between the different interfaces. In comparison to a decorator pattern, the adapter pattern only converts objects, while the decorator pattern adds new functionality to an existing object. Therefore, the decorator does not change the existing interface.

Why use OOAD opposed to sequential and functional.

-What is Data Encapsulation? Give a real world example

The process of binding data and corresponding methods (behavior) together into a single unit is called encapsulation in Java. ... Every Java class is an example of encapsulation because we write everything within the class only that binds variables and methods together and hides their complexity from other classes

In real life context, All the things that we have in our houses can be thought of as our data. Different type of people such as unknown visitors, friends and relatives come to our house. We don’t want to show each and every item of our house to each person hence we apply different access rules to them.

- Why use classes instead of modular programming?

The primary reason is so you can more closely associate behaviors with data. Now you can have a CoffeePot object and do whatever you want with it. Pass it around, extend its behavior, anything. The behaviors stay contained within the object.

- Can you do polymorphism in a single class? -Implement polymorphism without having inheritance?

Yes, using function overloading.

- Describe polymorphism in terms of objects in the interview room.

The person how is taking interview is son/daughter, employ and developer at the same time.

- What Architecture design is high in Cohesion.

(Answer: Service oriented Architecture)

Implement polymorphism without having interhitence?

(Answer: Explain function overloading.)

What is coupling and cohesion?

Which is better, coupling or cohesion? Why?

Cohesion refers to what the class (or module) can do. Low cohesion would mean that the class does a great variety of actions - it is broad, unfocused on what it should do. High cohesion means that the class is focused on what it should be doing, i.e. only methods relating to the intention of the class.

Example of Low Cohesion:

-------------------

| Staff |

-------------------

| checkEmail() |

| sendEmail() |

| emailValidate() |

| PrintLetter() |

-------------------

Example of High Cohesion:

----------------------------

| Staff |

----------------------------

| -salary |

| -emailAddr |

----------------------------

| setSalary(newSalary) |

| getSalary() |

| setEmailAddr(newEmail) |

| getEmailAddr() |

----------------------------

As for coupling, it refers to how related or dependent two classes/modules are toward each other. For low coupled classes, changing something major in one class should not affect the other. High coupling would make it difficult to change and maintain your code; since classes are closely knit together, making a change could require an entire system revamp.

Good software design has high cohesion and low coupling.

Diff b/w Polymorphism and Inheritance?

Inheritance is one in which a new class is created (derived class) that inherits the features from the already existing class(Base class). Whereas polymorphism is that which can be defined in multiple forms

Diff b/w Data Encapsulation and Data Abstraction?

Abstraction is the method of hiding the unwanted information. Whereas encapsulation is a method to hide the data in a single entity or unit along with a method to protect information from outside We can implement abstraction using abstract class and interfaces. Whereas encapsulation can be implemented using by access modifier i.e. private, protected and public.

How can we do Data Encapsulation?

Data Encapsulation is implemented by using access specifiers (Access Modifiers) and it defines the scope and visibility of a class member. And furthermore, we use getters and setters for member variable access.

How can we do Data Abstraction?

Using classes (interfaces and abstract classes) and header files.

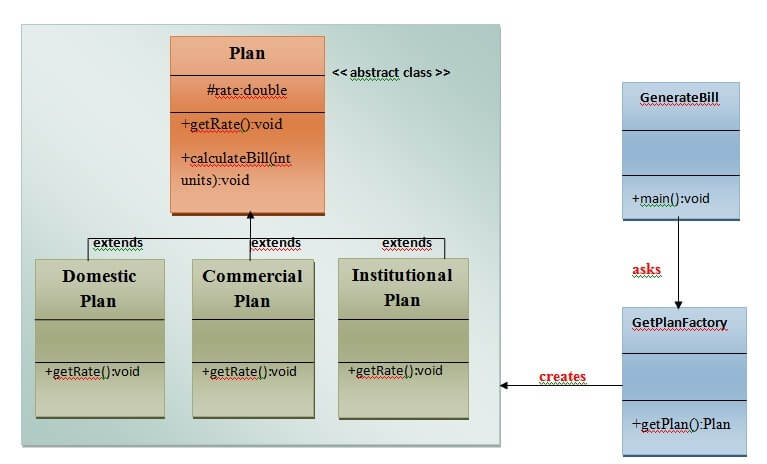
Does it is necessary that a class having Abstraction will have encapsulation?

In short no but if the abstract class/ interface is implemented and obj of child is created then encapsulation is done.

What is Factory pattern?

A Factory Pattern or Factory Method Pattern says that just define an interface or abstract class for creating an object but let the subclasses decide which class to instantiate. In other words, subclasses are responsible to create the instance of the class.

* We are going to create a Plan abstract class and concrete classes that extends the Plan abstract class. A factory class GetPlanFactory is defined as a next step.
* GenerateBill class will use GetPlanFactory to get a Plan object. It will pass information (DOMESTICPLAN / COMMERCIALPLAN / INSTITUTIONALPLAN) to GetPalnFactory to get the type of object it needs.



What is State pattern?

In State pattern a class behavior changes based on its state. This type of design pattern comes under behavior pattern.

Why do we use Design Patterns? What are the benefits?

Design patterns have two major benefits. First, they provide you with a way to solve issues related to software development using a proven solution. The solution facilitates the development of highly cohesive modules with minimal coupling. ... Second, design patterns make communication between designers more efficient.

What is difference b/w encryption and compression?

encryption is the process of encoding messages or information in such a way that only authorized parties can read it. Compression algorithms are techniques that exploit redundancy in data to reduce the size of the data representation.

Tell some encryption algos

## **AES, Twofish, Blowfish, RSA,** **Triple DES**

## **How would you encrypt any given string?**

## **What is open/closed principle**

## In object-oriented programming, the open–closed principle states "software entities should be open for extension, but closed for modification"; that is, such an entity can allow its behaviour to be extended without modifying its source code.

## **(SOLID:** [**S**ingle Responsibility Principle](https://stackify.com/solid-design-principles/) **,O**pen/Closed Principle, [**L**iskov Substitution Principle](https://stackify.com/solid-design-liskov-substitution-principle/), [**I**nterface Segregation Principle](https://stackify.com/interface-segregation-principle/), [**D**ependency Inversion](https://stackify.com/dependency-inversion-principle/)**)**

What is single responsibility principle

The single-responsibility principle is a computer-programming principle that states that every module, class or function in a computer program should have responsibility over a single part of that program's functionality, which it should encapsulate

what is Composition?

Composition vs inheritance

* nheritance: “is a.” E.g. The car is a vehicle. To make a long story short, when a child class inherits from a parent class, the child acquires all behaviors from the parent. Inheritance will make a class hierarchy — you can imagine it as a tree of classes. Inheritance is known as the tightest form of coupling
* Composition: “has a.” E.g. The car has a steering wheel. Composition is in contrast to inheritance, it enables the creation of complex types by combining objects (components) of other types, rather than inheriting from a base or parent class. To put it simply, composition contains instances of other classes that implement the desired functionality. Composition is a far looser coupling.

How can you reduce coupling?

Use infaces.

IWheel{

//wheel related methods

}

public class Ford{

IWheel wheel;

Ford(IWheel wheel){

this.wheel =wheel;// replaced by new JKWheel()

}

Or use static factory methods

public static IWheel createWheel(WHEEL wheel){

if(WHEEL.mrf.equals(wheel)

new MRFWheel(AirPressure pressure)

}

else if(WHEEL.JK.equals(wheel)

new JKWheel(AirPressure pressure)

}else{

New DumyWheel(AirPressure pressure)

}

calling:

new Ford(WheelFactory.createWheel(WHEEL.mrf)));

new Ford(WheelFactory.createWheel(WHEEL.JK)));

What are immutable strings

In java, string objects are immutable. Immutable simply means unmodifiable or unchangeable. Once string object is created its data or state can't be changed but a new string object is created.

Ask basic major concepts of ooad.

five core concepts of objects/classes, inheritance, encapsulation, methods, and polymorphism.

What are the advantages of object-oriented over functional programming?

Functional programming is having a stateless programming model. Object-oriented programming is having a stateful programming model.

Database queries on group by, order by, join and having clause.

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

Class B inherits from A. How will the stack look like with regards to A and B if you make an

object of B?

when the object of B is initialized then the default constructor is automatically called prior to B’s constructor so first A’s default constructor goes and after that B’s constructor goes. This all happens in reverse order.

When can you call static functions in a class?

You can simply call the static method of a class using class name followed by method name.

When can you call static functions in a class?

Lifetime of static variable in a class?

the lifetime of a static variable is the lifetime of the program.

What is scope resolution?

In C++, scope resolution operator is ::. It is used for following purposes.

1) To access a global variable when there is a local variable with same name

**2) To define a function outside a class.**

**3) To access a class’s static variables.**

**4) In case of multiple Inheritance:**

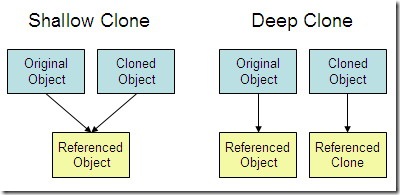
**5) For namespace**

**6) Refer to a class inside another class:**

Why is it better to have static variables in a class instead of global variables?

A static variable is only visible in the source file or {} block it is declared in, but exists and holds its value for the life of the program in global memory. A global variable is the same but visible anywhere. Included files and macros can also access a static variable when used where it is already visible.

Differentiate between shallow(copy by reference) and deep copy(copy by value).



What happens when an object is created?

Object memory is reserved in heap and values are initialized as null, refernce object is created in stack, variables are initialized either static or local, reference object is updated and map for memory allocation is created

What is difference between Query string and fragment?

<http://somesite.com/html/top.html#section_2>

The #section\_2 part is also called "the fragment". Its interpretation is up to the user agent (AKA client AKA browser), though the universal convention is that it's the anchor of an <a> tag to which the browser is asked to position itself.

http://server/path/program?query\_string#thefrag

The Query String, as explained [here](http://en.wikipedia.org/wiki/Query_string), is the part of the URL that's passed to the server-side program -- normally after a ? and up to but excluding the # if any. So, in:

the query string would be query\_string and the fragment would be thefrag.

What is Stateless Request?

Client send a request to the server and server replies accordingly. In the whole process It does not require the server to retain session information or a status about each communicating partner for multiple request.

## What is Composition?

Composition is a specialized form of aggregation. In composition, if the parent object is destroyed, then the child objects also cease to exist. Composition is actually a strong type of aggregation and is sometimes referred to as a “death” relationship. As an example, a house may be composed of one or more rooms. If the house is destroyed, then all of the rooms that are part of the house are also destroyed.

What is aggregation?

As an example, an employee may belong to one or more departments in an organization. However, if an employee’s department is deleted, the employee object would not be destroyed but would live on. Note that the relationships between objects participating in an aggregation cannot be reciprocal—i.e., a department may “own” an employee, but the employee does not own the department.

diffrence between heap and stack?

**Heap Allocation :**The memory is allocated during execution of instructions written by programmers. Note that the name heap has nothing to do with heap data structure. It is called heap because it is a pile of memory space available to programmers to allocated and de-allocate. If a programmer does not handle this memory well, [memory leak](https://www.geeksforgeeks.org/what-is-memory-leak-how-can-we-avoid/) can happen in the program.

**Stack Allocation :** The allocation happens on contiguous blocks of memory. We call it stack memory allocation because the allocation happens in function call stack. The size of memory to be allocated is known to compiler and whenever a function is called, its variables get memory allocated on the stack. And whenever the function call is over, the memory for the variables is deallocated. This all happens using some predefined routines in compiler. Programmer does not have to worry about memory allocation and deallocation of stack variables.

diffrence betwen implicit and explicit type casting?

 implicit is taken care of by the compiler itself, while explicit is done by the programmer

diffrence between parse and try parse?

Parse() method throws an exception if it cannot parse the value, whereas TryParse() method returns a bool indicating whether it succeeded. However, TryParse does not return the value, it returns a status code to indicate whether the parse succeeded and does not throw exception.

what is foreach loop and in which case it works fast?

Foreach loop is a control flow statement for traversing items in a collection. The execution time of .foreach() is dramatically affected by what happens inside each iteration. But when dealing with references foreach is faster.

difference between if else and switch?

In the case of 'if-else' statement, either the 'if' block or the 'else' block will be executed based on the condition. In the case of the 'switch' statement, one case after another will be executed until the break keyword is not found, or the default statement is executed.

what is for loop and in which case it works fine?

memory allocation for static members

static members occupy memory even if not even a single object is created. The don’t belong to any object as they are shared among them.

what is use of new keyword in method hiding?

When used as a declaration modifier, the new keyword explicitly hides a member that is inherited from a base class. When you hide an inherited member, the derived version of the member replaces the base class version. Although you can hide members without using the new modifier, you get a compiler warning. If you use new to explicitly hide a member, it suppresses this warning.

public class BaseC

{

public int x;

public void Invoke() { }

}

public class DerivedC : BaseC

{

new public void Invoke() { }

}

how to call base class method with child class object when both base and child class have implimentation of that method?

Java-🡪 super keyword.

C++

class left {

public:

void foo();

};

class bottom : public left, {

public:

void foo()

{

//base::foo();// ambiguous

left::foo();

// and when foo() is not called for 'this':

bottom b;

b.left::foo(); // calls b.foo() from 'left'

}

};

how one can say that overriding of method happens at run time and overloading at compile time?

Overloading🡪 compiler is aware of different functions with the same name.

Overriding🡪 compiler is unaware if the method is of parent or of child.

difference between class and object?

A class is a template for objects. ... A class also describes object behavior. An object is a member or an "instance" of a class. An object has a state in which all of its properties have values that you either explicitly define or that are defined by default settings.

difference between public ,private ,protected and internal keyword?

A public member is accessible from anywhere outside the class but within a program. You can set and get the value of public variables without any member.

A private member variable or function cannot be accessed, or even viewed from outside the class. Only the class and friend functions can access private members.

A protected member variable or function is very similar to a private member but it provided one additional benefit that they can be accessed in child classes which are called derived classes.

Internal It is used to specifies that access is limited to the current assembly.

what is inheritence ?single and multiple inheritence?

Inheritance 🡪 resuablilty of code.

Single🡪 one parent one child.

Multiple🡪 more the one parent and on child.

What is a virtual function?

A virtual function is a member function in the base class that we expect to redefine in derived classes.

Basically, a virtual function is used in the base class in order to ensure that the function is overridden. This especially applies to cases where a pointer of base class points to an object of a derived class.

class Base {

public:

virtual void print() {

// code

}

};

int main() {

Derived derived1;

Base\* base1 = &derived1;

// calls function of Base class

base1->print();

return 0;

}

What is function overloading?

function overloading or method overloading is the ability to create multiple functions of the same name with different implementations

What is a ternary operator?

The ternary operator is an operator that exists in some programming languages, which takes three operands rather than the typical one or two that most operators use. It provides a way to shorten a simple if else block. ... With this type of comparison, you can shorten the code using the ternary operator.

condition ? value\_if\_true : value\_if\_false

types of arguments that can be passed to a function?

 there are two types of arguments:

* Actual arguments
* Formal arguments

the two ways to pass arguments to the function:

* Pass by value
* Pass by reference

difference between out and ref keyword?

ref tells the compiler that the object is initialized before entering the function, while out tells the compiler that the object will be initialized inside the function.

The ref modifier means that:

1. The value is already set and
2. The method can read and modify it.

The out modifier means that:

1. The Value isn't set and can't be read by the method *until* it is set.
2. The method *must* set it before returning.

What is the super keyword?

The super keyword refers to superclass (parent) objects. It is used to call superclass methods, and to access the superclass constructor. The most common use of the super keyword is to eliminate the confusion between superclasses and subclasses that have methods with the same name.

what is the meaning of “IS-A” and “HAS-A” relationship?

An IS-A relationship is inheritance. On the other hand, HAS-A relationship is composition.

What is the diamond problem in inheritance?

The "diamond problem" (sometimes referred to as the "Deadly Diamond of Death") is an ambiguity that arises when two classes B and C inherit from A, and class D inherits from both B and C.

## Solution in java 8 Example

interface MyInterface1{

   public static int num = 100;

   public default void display() {

      System.out.println("display method of MyInterface1");

   }

}

interface MyInterface2{

   public static int num = 1000;

   public default void display() {

      System.out.println("display method of MyInterface2");

   }

}

public class InterfaceExample implements MyInterface1, MyInterface2{

   public void display() {

      MyInterface1.super.display();

      //or,

      MyInterface2.super.display();

   }

   public static void main(String args[]) {

      InterfaceExample obj = new InterfaceExample();

      obj.display();

   }

}

## Output

display method of MyInterface1

display method of MyInterface2

What is Inline function?

If a function is inline, the compiler places a copy of the code of that function at each point where the function is called at compile time.

Any change to an inline function could require all clients of the function to be recompiled because compiler would need to replace all the code once again otherwise it will continue with old functionality.

constructor overloading

We can have more than one constructor in a class with same name, as long as each has a different list of arguments.This concept is known as Constructor Overloading and is quite similar to [function overloading](https://www.geeksforgeeks.org/function-overloading-c/).

**What is difference between ArrayList and LinkedList ?**

Both the ArrayList and LinkedList classes implement the List interface, but they differ on the following features: • An ArrayList is an index based data structure backed by an Array. It provides random access to its elements with a performance equal to O(1). On the other hand, a LinkedList stores its data as list of elements and every element is linked to its previous and next element. In this case, the search operation for an element has execution time equal to O(n). • The Insertion, addition and removal operations of an element are faster in a LinkedList compared to an ArrayList, because there is no need of resizing an array or updating the index when an element is added in some arbitrary position inside the collection. • A LinkedList consumes more memory than an ArrayList, because every node in a LinkedList stores two references, one for its previous element and one for its next element.

**C# Interview Questions**

what are sealed classes

**Sealed classes** are used to restrict the users from inheriting the **class**. A **class** can be **sealed** by using the **sealed** keyword

how c# code is converted to machine code

The C# code is converted to MSIL, which is the assembly language that .NET's virtual machine understands. The MSIL is then compiled to machine code using Just-in-time compilation.

what is CLR

The Common Language Runtime, the virtual machine component of Microsoft .NET framework, manages the execution of .NET programs. Just-in-time compilation converts the managed code, into machine instructions which are then executed on the CPU of the computer.

what is JIT

Just-in-time manufacturing, also known as just-in-time production or the Toyota Production System, is a methodology aimed primarily at reducing times within the production system as well as response times from suppliers and to customers.

**Databases Interview Questions**