**write a java code to check if a number is prime?**

Public static boolean isPrime(int n){

if(n <= 1) return false;

for(int i = 2; i \* i < =n ;i++){

If(n % i == 0){

return false;

}

}

return true;

}

**What are method overloading and method overriding?**

**Overloading:**

Overloading occurs when two or more methods in one class have the same method name but different parameters. Like the constructors.

**Overriding:**

Overriding occurs when two methods have the same method name and parameters. One of the methods is in the parent class, and the other is in the child class. Overriding allows a child class to provide the specific implementation of a method that is already present in its parent class.

|  |  |
| --- | --- |
| Overloading | Overriding |
| The binding of overloaded method call to its definition has happens at compile-time | The binding of overridden method call to its definition happens at runtime |
| Static methods can be overloaded which means a class can have more than one static method of the same name | Static methods cannot be overridden |
| Method overloading is performed within class | Method overriding occurs in two classes that have IS-A (inheritance) relationships |
| Argument list should be different while doing method overloading | Argument list should be same in method Overriding |
| Return type of method does not matter in case of method overloading, it can be the same or different | In case of method overriding the overriding method can have more specific return type |

**What is OOP and fundamental concept of OOP?**

**OOP** stands for object oriented programming. It is a type of programming which is based on objects rather than functions and procedures. Related objects are working and communicating with each other to finish some business logical. Each object is an instance of a class. It has four main features which are abstraction, inheritance, encapsulation, and polymorphism.

**Encapsulation:**

Encapsulation refers to wrapping object state(member variables) and behavior(member methods) into a single unit. If you are creating a class, you are doing encapsulation. Encapsulation in java is achieved by using access modifier like private, public, protected and internal. If you want to expose some methods to a user, we could use public. If you want to hidden some methods to a user, we could use private. Encapsulation prevents access to implementation details.

**Abstraction**:

Abstraction is the concept of hiding the internal implementation and complex logic from a user. You can start a car by turning the key or pressing the start button. You don’t know how the engine is getting started. All the internal processes are hidden from the driver. There are two ways to achieve abstraction. One is abstract class and anther one is interface.

**Inheritance:**

Inheritance allows one class acquires the properties and functionalities of another class. The advantage of Inheritance is providing the reusability and extensibility of code and each subclass can defines its unique features, the rest of the features can be inherited from the parent class. The disadvantage of Inheritance is two classed are tightly coupled, so we could use composition instead of inheritance to get a loose coupling.

**Polymorphism**:

Polymorphism is the concept where an object behaves differently in different situations. There are two types of polymorphism. One is compile time polymorphism which is achieved by method overloading. It happens in the same class. Another one is runtime polymorphism which is achieved by method overriding. It happens in different classes. Example: employee, software, manager.

**Abstraction vs Encapsulation**

Abstraction is more about hiding the details at the design level, Encapsulation is more about hiding the details at implementation level. For example, we have a vehicle, we know it can move, it might move using tiers, it might fly, all we need to know is it can move. Then for encapsulation, for example, when we implement a HashMap, to solve the collision, we either use a linked list, or we use a red-black tree for each bucket. We can change the internal implementation without affecting the clients who use the HashMap.

**Difference between Abstract class and Interface?**

**Abstract class:**

A class that is declared with the abstract keyword is known as an abstract class in Java. It can have abstract and non-abstract methods (method with the body). All the abstract methods need to be implemented by child class. It can not be instantiated.

**Interface:**

The interface keyword is used to declare interface. It often provides a standard structure that the deriving classes would follow. Interfaces can contain only constants and abstract methods.Like abstract classes, Interfaces cannot be instantiated, they can only be implemented by classes or extended by other interfaces.

|  |  |
| --- | --- |
| Abstract class | Interface |
| Abstract class doesn't support multiple inheritance, it can extend only one class or one abstract class at a time | Interface supports multiple inheritance, it can extend any number of interfaces at a time |
| Abstract class can have final, non-final, static and non-static variables with any access specifier | Interface has only static, public and final variables |
| An abstract class can have both abstract and concrete methods | An interface can have only abstract methods. Since Java 8, it can have default and static methods also |
| The **abstract** keyword is used to declare abstract class | The **interface** keyword is used to declare interface |
| An abstract class can be **extended** using keyword "extends" | An interface class can be **implemented** using keyword "implements" |

**Difference between Static and no static variable or function in Java?**

Static keyword can be used with class, variable, method and block. Static members belong to the class instead of a specific instance, this means you can access it without an object. However, Non-static members belong to each individual instance of that class, this means you need to access it with an object.

**Static variable**: A static variable is common to all the instances (or objects) of the class because it is a class level variable.When we declare a member of a class as static, it means no matter how many objects of the class are created, there is only one copy of the static member.

**Non-static variable**: Non-static variable are specific to that instance of a class.

Access, bind, override.

**Static method:** A static method belongs to the class rather than the object of a class, it can be invoked without creating an instance of a class and can access static data member and static method. However, it cannot access non-static data members and method. Static method uses compile time or early binding.Static method cannot be overridden because of early binding.

**Non-static method:** the Non-method can access static data members and static methods as well as non-static data members and method. Non static method uses runtime or dynamic binding.Non-static method can be overridden because of runtime binding.

**Difference between Array and ArrayList**

C#

**Array** is a  fixed-size data structure which requires a size at the time of creation. Array contains same data type elements.

**ArrayList**is a dynamic-sized data structure which doesn’t require a specific size at the time of initialization. ArrayList allows different data type elements.

Java

An Array can contain both primitive data types or objects of a class depending on the definition of the array. But it has a fixed size.

An ArrayList can’t be created for primitive data types. It only contains an object. It has the ability to grow and shrink dynamically. An ArrayList implements the list interface, so it supports many additional operations like add, remove, indexof.

**How garbage collector works**

GC is an implementation of automatic memory management. The Garbage collector frees up space occupied by objects that are no longer in existence.

**What is finalize?**

Finalize as an object method used to free up unmanaged resources and cleanup before Garbage Collection(GC). It performs memory management tasks.

**What is sealed?**

Sealed class behave like a normal class, but you can not make a sealed class as a base class which means sealed class cannot be inherited. Sealed can not behave as base class but it can behave as derived class. Sealed class can not be abstract because abstraction forces to do the inheritance while sealed class can not be inherited.

Use Sql sever to establish a connection for a database. Class Sqlconnection is sealed by default.(System.Data.SqlClient)

When you use sealed modifiers on a method, it prevent the method to be overridden.

**What is delegate?**

C# delegates are similar to pointers to functions, in C or C++. A delegate is a reference type variable that holds the reference to a method. The reference can be changed at runtime.

Delegates are especially used for implementing events and the call-back methods. Delegate class is an agreement or contract between publisher and subscriber. It determines the signature of the event handler method in subscriber. All delegates are implicitly derived from the System.Delegate class.

**Difference between classes and structs?**

Structs can’t have destructors, but class can have destructors. Struct can’t inherit from another class where as a class can. Both structs and classes can inherit from an interface. A struct is value type where as a class is reference type. Struct can’t be used as base fro other structures or classes. Stuct can be instantiate without using the new operator.

**Difference between virtual method and abstract method?**

Virtual methods have an implementation and provide the derived classes with the option of overriding it. Abstract methods do not provide an implementation and force the derived classes to override the method.

**Constructors and Destructors?**

We use constructor to create an object with using new keyword. It has exactly the same as class name. It has no return type and can be overloaded.

Garbage collector will call destrcutor to free up space on heap memory when an object goes out of scope. Memory will be released before finishing the program. Destrcutor has the same name as class name with a prefixed tilde. It can not be overloaded.

**What is Generic?**

Generics is like placeholder which allow us to design classed and methods decoupled from the data types. If I want to design a container, let’s a say list. I don’t know what type of elements will be add in the list by different users. I don’t want to write a specific list for a specific data type, like int list, string list, etc. That’s why we need to use generic to increase reusability of the code.

**What is Singleton?**

Singleton is a creational design pattern. it will ensure that there is only one instance of a class is created. Instance of singleton class should be globally accessible so that each class can use it. For example, the database instance could be singleton, so it can be accessed by different client objects. When we implement a singleton class, we should set constructor private. This will prevent other classes to instantiate the singleton class. Then we will declare a static variable to store singleton instance. Finally, we will set getInstance method to be public static, so it can provide a global point of assess to singleton object. We also have different type of singleton, like early initialization and lazy initialization. We also need to pay attention to the synchronized problem in singleton class.

**Dependency Injection?**

The main purpose of dependency injection is to make object loosely coupled with each other. This could increase code flexibility and maintainability. In the spring frame work, we will create instances in the spring container. We call these instances spring beans. Once in our classes, we need to use these instance. We will ask spring container do you have the instance I want. Then the spring container will inject these instance to your code. You can create the beans by using annotation or we could set configuration file like Xtml. There are there ways to inject beans to our code. We can use constructor, setter method and annotation.

What is encapsulation

Static keyword

Design Patterns

#### What is early and late binding? What is a copy constructor? What is static and dynamic binding?

#### Difference between Array and ArrayList