Oops

[Object Oriented Programming]

What are OOPs?

Oops is a computer programming model that organizes software design around data, or objects, rather than functions and logic.

What are the main features/ pillars of OOPs?

1. **Inheritance**: It allows one class to inherit the properties and behaviors of another class. The class that is inherited from is called the "parent" or "superclass," and the class that inherits is called the child or subclass.
2. **Encapsulation**: A class is an example of encapsulation that consists of data and methods that have been bundled into a single unit. It prevents classes from accessing and changing fields and methods of a class from the outer side. This also helps to achieve data hiding.
3. **Polymorphism**: It is a concept by which we can perform a single action in different ways.
4. **Abstraction**: It is the concept of object-oriented programming that shows only essential attributes and hides unnecessary information.

What are the types of Inheritance in Java?

1. **Single Inheritance**: It means A class can inherit from only one superclass.
2. **Multilevel Inheritance**: When A class is derived from another derived class, creating a multilevel chain of inheritance.
3. **Hierarchical Inheritance**: When Multiple classes inherit from a single superclass.
4. **Multiple Inheritance**: Java does not support multiple inheritance with classes (to avoid the diamond problem), but it allows a class to implement multiple interfaces. This way, a class can inherit behavior from multiple sources.
5. **Hybrid Inheritance**: This is a combination of two or more types of inheritance. While Java does not support hybrid inheritance directly with classes due to the restriction on multiple inheritance, it can be achieved using interfaces.

What are the types of Encapsulation in Java?

1. **Data hiding**: It is a process of hiding unwanted information, such as restricting access to any member of an object.
2. **Data binding:** It is a process that creates a connection between the application's UI and the data. When the data changes its value, the UI elements that are bound to the data will also change.

What are the types of Polymorphism in Java?

We can achieve polymorphism using Method overriding, Method overloading

- Static /compile time polymorphism: It could be achieved through overloading.
- Dynamic /runtime polymorphism: It could be achieved through overriding.

What is Overloading and Overriding?

Overloading: When using the same method name for various implementations, differentiated by parameters.

Overriding: It allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its superclass or parent class.

What is meant by Structured Programming?

Structured Programming is a programming paradigm aimed at improving the clarity, quality, and development time of software by using a clear, linear, and structured flow of control.

What are some advantages of using OOPs?

- OOPs are very helpful in solving very complex level of problems.
- Highly complex programs can be created, handled, and maintained easily using object-oriented programming.
- OOPs, promote code reuse, thereby reducing redundancy.
- OOPs also help to hide unnecessary details with the help of Data Abstraction.

What is a Constructor?

A constructor is a special method used to initialize objects. It is called when an object of a class is created.

What are access modifiers?

Access modifiers are keywords that set the accessibility of classes, methods, and other members.

- private: Accessible only within the class.
- default (package-private): Accessible only within the same package.
- protected: Accessible within the same package and subclasses.
- public: Accessible from any other class.

Abstract

A keyword used in a class or method definition, which specifies that the method/class is not going to be instantiated, but should be inherited by other methods or classes

Coupling

It refers to how strongly a software element is connected to other elements.

Cohesion

It is defined as used to perform specialized tasks (single task) instead of multiple tasks with a single Java class.

Association

It is a connection or relation between two separate classes that are set up through their objects. It can be one-to-one, one-to-many, many-to-one and many-to-many.

Composition and aggregation are two types of association.

1. **Aggregation:** It implies a relationship where the child can exist independently of the parent. For example, bank and employee, delete the bank, and the employee still exists. Whereas
2. **Composition:** It implies a relationship where the child cannot exist independently of the parent.

Methods

What is a static method?

Static methods belong to the class rather than any instance of the class. They are called on the class itself.

What is the final method?

A method that cannot be overridden by subclasses.

What are Instance Methods?

Instance Methods belong to an instance of a class. They can access instance variables and instance methods directly.

What are Abstract Methods?

Abstract Methods are declared without an implementation. They must be implemented by subclasses and can only be declared in abstract classes or interfaces.

What are Synchronized Methods?

Synchronized Methods are used to control access to a method by multiple threads.Only one thread can execute a synchronized method at a time on the same object.

What are Native methods?

Native methods can access system-specific functions and APIs that are not available directly in Java.

What is Synchronization?

Synchronization is the capability to control the access of multiple threads to any shared resource.

What is Asynchronization?

Asynchronization refers to the execution of main program flow tasks independently. It allows a program can continue executing other tasks while waiting for a particular task to complete.

Keywords

Abstract	Used to declare a class or method that cannot be instantiated or must be overridden.
Assert	Used for debugging purposes to make an assertion.
Final	A non-access modifier is used for classes, attributes, and methods, which makes them non-changeable.
Static	This is used for memory management mainly. We can apply static keywords with variables, methods, blocks, and nested classes.
Synchronized	This ensures that only one thread can access a shared resource at a time

This	Refers to the current invoking object in a method or constructor.
Super	Refers to the superclass of the current object.
Transient	Prevents serialization of a field.
Volatile	Indicates that a variable may be changed unexpectedly by other parts of the program.
new	Used to create a new instance of a class.
instance of	Check whether an object is an instance of a specific class or subclass.
native	It is applied to a method to indicate that the method is implemented in native code using JNI (Java Native Interface).