How will you answer this question in a Java interview?

What is Object Oriented Programming?

- Object-oriented programming (OOP) is a programming paradigm or a problem solving technique[1]
- Object-oriented programming maps the programming model to real world concepts. This is technique to think real world in terms of objects[1]
- An object-oriented program may be viewed as a collection of interacting objects, as opposed to the conventional model, in which a program is seen as a list of tasks (subroutines) to perform[1]
- In OOP, each object is capable of receiving messages, processing data, and sending messages to other objects[1]
- Object-Oriented Programming is a methodology or paradigm[2]

What are the advantages of object oriented software development?

- Modular development of code, which leads to easy maintenance and modification.
- Reusability of code[3]
- Improved reliability and flexibility of code[3]
- Increased understanding of code[3]
- Highly cohesive (well-integrated) and loosely coupled code-small changes don't break your code somewhere else[4]
- Ease in software design as you could think in the problem space rather than the machine's bits and bytes. You are dealing with high-level concepts and abstractions. Ease in design leads to more productive software development[5]
- Ease in software maintenance: object-oriented software are easier to understand, therefore easier to test, debug, and maintain[5]
- Reusable software: you don't need to keep re-inventing the wheel and re-write the same functions for different situations. The fastest and safest way of developing a new application is to reuse existing codes - fully tested and proven codes[5]

What are the features of Object-Oriented Programming?[3]

- Encapsulation
- Inheritance
- Polymorphism
- Abstraction

What is Encapsultaton and how is it implemented in Java?[3]

Encapsulation provides objects with the ability to hide their internal characteristics and behavior. Each object provides a number of methods, which can be accessed by other objects and change its internal data. In Java, there are three access modifiers: **public, private and protected**. Each modifier imposes different access rights to other classes, either in the same or in external packages. Some of the advantages of using encapsulation are listed below:

- The internal state of every objected is protected by hiding its attributes.
- It increases usability and maintenance of code, because the behavior of an object can be independently changed or extended.
- It improves modularity by preventing objects from interacting with each other, in an undesired way.

A java class is the example of encapsulation.[2]

Polymorphism

Polymorphism is the ability of programming languages to present the same interface for differing underlying data types. A polymorphic type is a type whose operations can also be applied to values of some other type.[3]

In java, we use method overloading and method overriding to achieve polymorphism.[2]

Inheritance

Inheritance provides an object with the ability to acquire the fields and methods of another class, called base class. Inheritance provides re-usability of code and can be used to add additional features to an existing class, without modifying it. Implemented using the extends or implements keywords.[3]

It is used to achieve runtime polymorphism.[2]

Abstraction

Abstraction is the process of separating ideas from specific instances and thus, develop classes in terms of their own functionality, instead of their implementation details. Java supports the creation and existence of abstract classes that expose interfaces, without including the actual implementation of all methods. The abstraction technique aims to separate the implementation details of a class from its behavior.[3]

In java, we use abstract class and interface to achieve abstraction.[2]

Referencces:

1. http://a4academics.com/interview-questions/54-core-java-swing/412-oops-and-core-java-interview-questions-with-answers
2. http://www.javatpoint.com/java-oops-concepts
3. https://www.javacodegeeks.com/2014/04/java-interview-questions-and-answers.html
4. http://javarevisited.blogspot.com/2012/03/10-object-oriented-design-principles.html
5. https://www.ntu.edu.sg/home/ehchua/programming/java/J3a_OOPBasics.html