Object-Oriented Programming (OOP) is a programming approach that models everything as objects, unlike procedural programming. Java was designed to follow this pattern, making it an object-oriented programming language.

A class is a blueprint for creating an object. To represent attributes or properties, classes use instance variables, and for behaviors or actions, it uses instance methods. The process of creating objects from a class is called instantiation. Objects are real entities with attributes and behaviors defined by their models, such as a circle, dog, fish, human, etc. Once a template for an object has been created, creating similar objects becomes easier.

Abstraction is the process of hiding the low-level details of an object and exposing only useful attributes to the user. In abstraction, the user only knows what a function (method in Java) does but not how it does it. Abstraction can be implemented in Java using either the non-access modifier abstract keyword or interface. Encapsulation is the process of encapsulating and hiding the details of a class implementation from the user. Encapsulation makes for the security of an application as the user of a program cannot access the private variables. They can only do so using its public methods, usually accessor and mutator methods.

To implement encapsulation in Java, class attributes should be declred as private, and public setter and getter methods provided. In conclusion, this article covered the basics of OOP and how to create objects, classes, abstraction, and encapsulation in Java. With this knowledge, beginners can progress to advanced topics like inheritance and polymorphism, which build upon these fundamental concepts.