

Class: A class is a template or blueprint or prototype that defines data members

Object: Any entity that has state and behavior is known as an object.

Constructor:

in java is used to create the instance of the class.

Inheritance :

When one object acquires all the properties and behaviors of a parent object, it is known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

polymorphism :

If one task is performed in different ways, it is known as polymorphism. For example: to convince the customer differently, to draw something, for example, shape, triangle, rectangle, etc.

Compile-time :

polymorphism allows us to use many methods with the same name but differing signatures and return types.

Run-time polymorphism :

is associated with different classes, but it allows us to use the same method with different signature names.

Abstraction :

Hiding internal details and showing functionality is known as abstraction. For example phone call, we don't know the internal processing.

Encapsulation:

Binding (or wrapping) code and data together into a single unit are known as encapsulation. For example, a capsule, it is wrapped with different medicines.

Access Specifier:

Public Private Protected Default

Method Overloading :

If a class has multiple methods having same name but different in parameters, it is known as

static :The static variable can be used to refer to the common property of all objects.

Method Overriding:

If subclass (child class) has the same method as declared in the parent class, it is known as method overriding in Java.

@Autowired:

@Autowired annotation, the spring container auto-wires the bean by matching data-type.

@SpringBootApplication :

used to mark a configuration class that declares one or more @Bean methods and also triggers auto-configuration and component scanning

@ComponentScan:

It is used when we want to scan a package for beans.

@Bean:It tells the method to produce a bean to be managed by Spring Container.

@Controller:

This is simply a specialization of the @Component class,

@RestController:

It's a convenient annotation that combines @Controller and @ResponseBody,

@ResponseBody: with the It will every request handling method of the controller class.

@Enableswagger :

Swagger2 is an open source project used to generate the REST API documents for RESTful web services.

@Service:

Spring that class contains the business logic.

@Repository:

The repository does all the operations related to the database.

@SpringBootApplication:

@EnableAutoConfiguration, @ComponentScan, and @Configuration.

@EnableAutoConfiguration: It auto-configures the bean that is present in the classpath and configures it to run the methods.

@GetMapping: It maps the HTTP GET requests

@RequestBody:@RequestBody, the Spring framework binds the incoming HTTP request body to that parameter.

@ResponseBody:It binds the method return value to the response body.