* Spring is a loosely coupled.
* Spring is a light-weight application.
* Spring works mostly on HAS-A relationship.
* Spring has a IOC container (inversion of control):

It will creates a Bean, it will configures the bean and manages the bean till it usage.

It helps to read data from configuration file/ property files.

IOC container helps in achieving Dependency injection.

* Dependency Injection:

A 🡺 B 🡺 C 🡺 D 🡺 tightly coupled

It helps to achieve loosely coupling.

It is of 2 types

1. With setter injection
2. With Constructor injection

* Bean:

It is an object, by default it is singleton object.

To create a bean we will use @Bean.

It is method level.

* Bean Factory: it is group of beans combined.

.jar 🡺 group of .class files

.war 🡺 group of .jar files

.ear 🡺 group of .war files

Java 🡺 .jar

Maven 🡺 dependency

Diagram

Description automatically generated

JDBC: configurations, Queries SQL

ORM tool(JPA)🡺 Object Relational Mapping 🡺 implement by Hibernate

* Spring boot is autoconfigurer.
* Spring boot is handle dependency management.
* Spring boot has embedded Tomcat and maven.
* Spring boot has actuators for monitoring purpose.

@EnableAutoconfiguration 🡺

@SpringbootConfiguration

@ComponentScan

@SpringbootApplication

@RequestMapping(“/student/nb/”)

@Controller

@ResponseBody 🡺 it will converts java to json and sends to HttpResponse

@RequestBody 🡺 HttpRequest 🡺 It will converts Json to Java

{

“name”:”ABC”,

}

@RestController 🡺 @Controller + @ResponseBody

@Service

@Repository

@Transactional

@Autowired

@Bean

@Component

@Qualifier

@Configuration

@PostMapping 🡺 Save

@GetMapping 🡺 Fetch

@PutMapping 🡺 update

@DeleteMapping 🡺 delete

@PathVariable 🡺 reading data from URL

@RequestParam 🡺 reading data as keyvalue pair

@ControllerAdvice 🡺 exception

@RestControllerAdvice 🡺 exceptions

@ExceptionalHandler 🡺 exceptions