

# Java and Spring Boot Interview Comparison Questions

## Spring Framework Core

### Comparison Question

@Component vs @Bean  
@Controller vs @RestController  
@RequestParam vs @PathVariable  
@RequestParam vs @RequestBody  
@Autowired vs @Resource vs @Inject  
ApplicationContext vs BeanFactory  
@Configuration vs @Component  
Constructor Injection vs Setter Injection vs Field Injection  
@Service vs @Repository vs @Component  
@RequestMapping vs @GetMapping/@PostMapping  
@Valid vs @Validated  
@Qualifier vs @Primary  
@Scope("prototype") vs @Scope("singleton")  
@EnableAsync vs @Async  
@EventListener vs @TransactionalEventListener  
@Profile vs @Conditional  
@DependsOn vs @Order  
@Import vs @ComponentScan  
XML Configuration vs Annotation Configuration vs Java Configuration  
FactoryBean vs @Bean  
@PostConstruct vs @PreDestroy vs InitializingBean vs DisposableBean  
BeanPostProcessor vs BeanFactoryPostProcessor

## Spring Boot

### Comparison Question

Spring vs Spring Boot  
@SpringBootApplication vs @EnableAutoConfiguration + @ComponentScan + @Configuration  
application.properties vs application.yml  
@ConfigurationProperties vs @Value  
Embedded Server vs External Server  
spring-boot-starter vs Traditional Dependencies  
@SpringBootTest vs @WebMvcTest vs @DataJpaTest  
CommandLineRunner vs ApplicationRunner  
@EnableAutoConfiguration vs @Conditional  
Spring Boot Actuator vs Custom Monitoring  
DevTools vs Manual Restart  
Fat JAR vs WAR Deployment  
application.properties vs bootstrap.properties  
@RefreshScope vs Static Configuration

# Java Core - Language Features

## Comparison Question

Abstract Class vs Interface  
Interface (Java 7) vs Interface (Java 8+)  
Checked Exception vs Unchecked Exception  
throw vs throws  
final vs finally vs finalize  
== vs .equals()  
String vs StringBuilder vs StringBuffer  
pass-by-value vs pass-by-reference in Java  
static vs instance methods  
static vs non-static nested classes  
Inner Class vs Static Nested Class vs Local Class vs Anonymous Class  
Overloading vs Overriding  
Compile-time Polymorphism vs Runtime Polymorphism  
this vs super  
package-private vs protected vs public vs private  
transient vs volatile  
Serializable vs Externalizable  
shallow copy vs deep copy  
Heap vs Stack Memory  
PermGen vs Metaspace  
Young Generation vs Old Generation

# Java Collections

## Comparison Question

ArrayList vs LinkedList  
ArrayList vs Vector  
HashSet vs LinkedHashSet vs TreeSet  
HashMap vs HashTable vs ConcurrentHashMap  
HashMap vs TreeMap vs LinkedHashMap  
HashSet vs HashMap  
Comparable vs Comparator  
Iterator vs ListIterator vs Spliterator  
Enumeration vs Iterator  
Iterator vs forEach  
Collection vs Collections  
List vs Set vs Queue  
Queue vs Deque  
PriorityQueue vs TreeSet  
WeakHashMap vs HashMap  
IdentityHashMap vs HashMap  
Fail-Fast vs Fail-Safe Iterators  
Arrays.asList() vs new ArrayList()  
Collections.synchronizedList() vs CopyOnWriteArrayList  
Collections.unmodifiableList() vs ImmutableList

# Java Concurrency

## Comparison Question

Process vs Thread  
User Thread vs Daemon Thread  
wait() vs sleep()  
wait() vs park()  
notify() vs notifyAll()  
Runnable vs Callable  
Future vs CompletableFuture  
execute() vs submit() in ExecutorService  
synchronized method vs synchronized block  
volatile vs synchronized  
volatile vs Atomic classes  
CountDownLatch vs CyclicBarrier  
Semaphore vs Mutex  
ReentrantLock vs synchronized  
ReadWriteLock vs ReentrantLock  
ThreadLocal vs Instance Variables  
Fork/Join vs ExecutorService  
parallelStream() vs stream()  
Thread.start() vs Thread.run()  
interrupted() vs isInterrupted()  
ThreadPoolExecutor vs ForkJoinPool  
FixedThreadPool vs CachedThreadPool vs ScheduledThreadPool

# Java 8+ Features

## Comparison Question

Stream vs Collection  
map() vs flatMap()  
Stream vs Parallel Stream  
Optional.of() vs Optional.ofNullable()  
findFirst() vs findAny()  
Predicate vs Function vs Consumer vs Supplier  
Method Reference vs Lambda Expression  
Default Methods vs Abstract Methods  
Stream.forEach() vs Iterable.forEach()  
reduce() vs collect()  
Intermediate Operations vs Terminal Operations  
anyMatch() vs allMatch() vs noneMatch()

# JPA/Hibernate/Database

## Comparison Question

JPA vs Hibernate  
JPA vs JDBC  
@OneToMany vs @ManyToOne  
@ManyToMany vs Two @OneToMany  
FetchType.LAZY vs FetchType.EAGER  
@JoinColumn vs @JoinTable  
First Level Cache vs Second Level Cache  
get() vs load() in Hibernate  
save() vs persist() vs saveOrUpdate()  
merge() vs update()  
JPQL vs Native SQL vs Criteria API  
@Entity vs @Table  
@Id vs @EmbeddedId vs @IdClass  
@GeneratedValue strategies comparison  
@Transactional(readOnly=true) vs @Transactional(readOnly=false)  
Optimistic Locking vs Pessimistic Locking  
@Version vs @Lock  
CascadeType.ALL vs CascadeType.PERSIST vs CascadeType.MERGE  
orphanRemoval vs CascadeType.REMOVE  
@Embeddable vs @Entity  
@ElementCollection vs @OneToMany  
@Inheritance strategies comparison  
Named Query vs Dynamic Query  
EntityManager vs Session  
JpaRepository vs CrudRepository vs PagingAndSortingRepository  
@Modifying vs Regular Query  
@Query vs Query Methods vs Specifications

# Spring Security

## Comparison Question

Authentication vs Authorization  
@PreAuthorize vs @PostAuthorize  
@Secured vs @RolesAllowed vs @PreAuthorize  
Form-based Authentication vs Token-based Authentication  
Session-based vs Stateless Security  
OAuth vs OAuth2  
JWT vs Session Tokens  
CSRF vs XSS  
@EnableWebSecurity vs @EnableGlobalMethodSecurity  
antMatchers() vs mvcMatchers()  
hasRole() vs hasAuthority()  
PasswordEncoder vs plain text passwords  
Remember-me vs Persistent Login  
HTTP Basic vs HTTP Digest Authentication

# Spring Web/REST

## Comparison Question

REST vs SOAP  
REST vs GraphQL  
@RequestBody vs @ResponseBody  
@RequestBody vs @ModelAttribute  
ResponseEntity vs @ResponseBody  
PUT vs POST vs PATCH  
GET vs POST  
@RestController vs @Controller + @ResponseBody  
@ExceptionHandler vs @ControllerAdvice  
HandlerInterceptor vs Filter  
OncePerRequestFilter vs GenericFilterBean  
Servlet vs Filter  
Session vs Token Authentication  
Cookie vs Session  
Stateful vs Stateless  
Content Negotiation vs Fixed Response Format  
@CrossOrigin vs CORS Filter  
Forward vs Redirect  
RequestDispatcher.forward() vs HttpServletResponse.sendRedirect()  
WebMvcConfigurer vs WebMvcConfigurerAdapter

# Spring AOP

## Comparison Question

@Before vs @After vs @Around  
@AfterReturning vs @AfterThrowing  
JDK Dynamic Proxy vs CGLIB Proxy  
Compile-time Weaving vs Load-time Weaving vs Runtime Weaving  
@Aspect vs @Component  
Pointcut vs Join Point vs Advice  
AOP Proxy vs Target Object  
Spring AOP vs AspectJ  
Method-level AOP vs Class-level AOP

# Testing

## Comparison Question

@Mock vs @MockBean  
@Mock vs @Spy  
@MockBean vs @SpyBean  
@WebMvcTest vs @SpringBootTest  
@DataJpaTest vs @SpringBootTest  
Unit Testing vs Integration Testing vs End-to-End Testing  
TDD vs BDD  
JUnit vs TestNG  
MockMvc vs RestTemplate vs TestRestTemplate  
@BeforeEach vs @BeforeAll  
@Test vs @ParameterizedTest  
@DisplayName vs Method Name  
Mockito.when() vs Mockito.doReturn()  
verify() vs assert()  
@DirtiesContext vs Test Isolation  
H2 vs Test Containers  
@Sql vs @DataJpaTest

# Microservices & Distributed Systems

## Comparison Question

Monolithic vs Microservices  
RestTemplate vs WebClient vs FeignClient  
Synchronous vs Asynchronous Communication  
HTTP vs Message Queue Communication  
API Gateway vs Load Balancer  
Service Discovery vs Load Balancing  
Client-side Load Balancing vs Server-side Load Balancing  
Eureka vs Consul vs Zookeeper  
Circuit Breaker vs Retry  
Hystrix vs Resilience4j  
Saga Pattern vs Two-Phase Commit  
Event Sourcing vs Traditional State Storage  
CQRS vs Traditional Architecture  
API Gateway vs BFF (Backend for Frontend)  
Docker vs Virtual Machine  
Kubernetes vs Docker Swarm  
ConfigMap vs Secret in Kubernetes  
Horizontal Scaling vs Vertical Scaling  
Stateful vs Stateless Services  
Sidecar vs Library Pattern

# Design Patterns

## Comparison Question

Factory Pattern vs Abstract Factory Pattern  
Factory Method vs Simple Factory  
Builder Pattern vs Constructor  
Singleton vs Static Class  
Adapter Pattern vs Decorator Pattern  
Adapter vs Facade vs Proxy  
Strategy Pattern vs State Pattern  
Observer Pattern vs Pub-Sub Pattern  
Template Method vs Strategy Pattern  
Composite Pattern vs Decorator Pattern  
Chain of Responsibility vs Command Pattern  
Iterator Pattern vs For-Each Loop  
Flyweight vs Singleton  
Bridge Pattern vs Adapter Pattern  
Mediator vs Observer  
Prototype vs Factory Pattern  
Command Pattern vs Strategy Pattern

# Spring Reactive

## Comparison Question

Spring MVC vs Spring WebFlux  
Mono vs Flux  
map() vs flatMap() in Reactor  
subscribe() vs block()  
Reactive Streams vs Java Streams  
Project Reactor vs RxJava  
@Controller vs @RestController in WebFlux  
RouterFunction vs @RequestMapping  
Backpressure vs Throttling  
Cold Publisher vs Hot Publisher  
publishOn() vs subscribeOn()

# Caching

## Comparison Question

@Cacheable vs @CachePut  
@CacheEvict vs @CachePut  
Local Cache vs Distributed Cache  
Ehcache vs Hazelcast vs Redis  
Cache-aside vs Read-through vs Write-through vs Write-behind  
TTL vs TTI  
LRU vs LFU vs FIFO Eviction  
Spring Cache vs JCache

# Message Queuing

## Comparison Question

JMS vs AMQP  
RabbitMQ vs Kafka  
Queue vs Topic  
Point-to-Point vs Publish-Subscribe  
Push Model vs Pull Model  
At-least-once vs At-most-once vs Exactly-once Delivery  
Persistent vs Non-persistent Messages  
Synchronous vs Asynchronous Messaging

# Build Tools & Dependency Management

## Comparison Question

Maven vs Gradle  
compile vs runtime vs provided scope in Maven  
implementation vs api in Gradle  
Maven Repository vs Local Repository  
SNAPSHOT vs RELEASE versions  
Multi-module vs Single-module Project  
BOM vs Direct Dependency Management

# Logging & Monitoring

## Comparison Question

SLF4J vs Log4j vs Logback  
ERROR vs WARN vs INFO vs DEBUG vs TRACE  
Console Appender vs File Appender  
Synchronous Logging vs Asynchronous Logging  
MDC vs ThreadLocal  
Metrics vs Logs vs Traces  
Prometheus vs Graphite  
Pull-based vs Push-based Monitoring

# Performance & Optimization

## Comparison Question

Eager Initialization vs Lazy Initialization  
Connection Pooling vs Creating Connections  
Database Index vs Full Table Scan  
Caching vs Recomputing  
Vertical Scaling vs Horizontal Scaling  
Synchronous vs Asynchronous Processing  
Batch Processing vs Real-time Processing  
N+1 Query Problem vs Join Fetching

# Cloud & Deployment

## Comparison Question

IaaS vs PaaS vs SaaS

Public Cloud vs Private Cloud vs Hybrid Cloud

AWS vs Azure vs GCP

EC2 vs Lambda

Container vs Serverless

Blue-Green Deployment vs Canary Deployment

Rolling Deployment vs Recreate Deployment

Terraform vs CloudFormation

---

*Note: These are comparison questions only. The actual interview would require detailed understanding of the differences, use cases, advantages, and disadvantages of each concept.*