Exception Handlling and Stream API in java?

1. **What are the type of Errors in java?**
   a. compile time error
   b. Runtime Error
   c. Logical Error(bugs)
2. What is expetions in java?
   a. exceptions are runtime errors occurs in java application, for example if you have handlled the java null cases in your application at the runtime the cases will throw the runtime error which is nullpointerexceptions
3. Exception Hierarchy in java
   a. Object(Every class in java extends the object class)
   b. Throwable
   c. Exception
      i. Runtime Exceptions - unchecked Exceptions
      ii. SQL Exceptions - checked exceptions
      iii. Io exceptios - unchecked exceptions
4. Throw key word in java?
   a. throw key word is use to throw the exception,
5. Custom exceptions in java?
   a. Custom exceptions are those which are make by the programmer according to its requirements.
   b. To generate custom exceptions you have to create your custom exception class which is extending the exception or runtime exceptions
6. Throws Keyword in Java?
   a. Throws keyword is used with methods to handle the exceptions, if you have a method which contains some critical exceptios and you dont wants to handle it in your method, then simply use throws expetion in method, and whatever the class which using that method will handle that exception with try catch
7. Try with Finally block?
   a. Finally block in java is used to execute the important code, such as closing resources, regardless of whether an exception occurs or not. It always run after try and catch blocks, ensuring that cleanup operations,

Stream API Java

Javaquetions-

1. what is constructor in java
   a. we can define constructor in java as special type of method, which are used for declare defaults some default values.
   b. whenever we creats an object of an class a default constructor of class is called.
   c. constructor have same name as its class name and there is no return type
2. What is parametersised constructor.
   a. in parameterised counstructor user can pass the defult values at the time of creating object.
3. What is anonymous object and reference object
   a. anonymous object defined as the class without reference
   b. and the class with the reference called as reference class
4. This keyword in java
   a. This keyword specially used to avoid name conflict between the instance variable and the method parameter of the class.
   b. In java we have instance variable and local variable, when i create a class with private instance variable in that class, we need getters and setters method to
   c. if the names of instance variable and the local variables are same then this key word comes in to the picture,
5. Can we print something on the console without using a main method in java?
   a. yes we can print without using a main method using static block
6. What is finally block in java?
   a. Finally block in java is used to execute the important code in your method, irespect of there is exception or not in your method, finally block will always exe
7. Can we have multiple finally block in java?
   a. NO we can not have multiple finally block. There is only one finally for each try catch pair
8. Which collection have you used in your project
9. what is the difference between the collections
10. Can we include a class as key in hashmap?
    a. yes we can include a class as key in hashmap
11. Do you know what is reflection in java?
    a. The reflection in nothing but a feature which allows us the change the behaviour of the method
12. Do you aware about the java8 Features
13. What is the functional Interface
    a. Functional interface are the one interface which contains the only single abstract method
14. What is the difference between the comparable and compatable
15. Can functional interface extend another interface?
    a. Functional interface can not extend another interface, because we know the functional interface contains only single abstract method and if the another interface contains one more abstract method then the rule will breaks
16. what is steam api, are you aware of stream api
17. what is the difference between the filter and map
    a. we can say that map basically operates on a every element and it returns entire element list as it is but if we wants filtered out or if we wants to disgards based on conditon, in filter we reduced elements from the list

Java spring boot interview quetions

1. **Which are the recent project that you are worked on**

   Recent project which i worked on

   a. **Invoice Discounting**: I developed a discounting platform for program managers, automating the process that was previously managed using Excel. When a sales invoice is generated for a customer, an API triggers the entry into my discounting platform, allowing program managers to proceed with the discounting process seamlessly.
   b. **Cogoscore Calculation for Credit Limits**: For Cogoport, I created a system to assign credit limits to customers based on specific credit terms. I fetched valid credit terms from third-party sources, compared them with our master credit terms and cogoscore data, and determined the final cogoscore for the customer. The customer's credit limit is then assigned according to this cogoscore.
   c. **Borrower CRM**: I developed a Customer Relationship Management (CRM) system to manage and display customer information for those using our fintech portal for cogoscore calculation and discounting.
   d.
   e. **Onboarding CRM**: I built a system to manage onboarding information and KYC verification for customers accessing our fintech portal for cogoscore calculation and discounting.

f. **Dashboards**: I designed dashboards for analyzing key metrics, including the total credit limits assigned and the amount of discounting completed.
g. Cogoscore Calculation For Export Factoring -

1. **How can we optimise queries**
   a. Proper indexing of database tables can significantly enhance query performance
   b. Reveiw your sql queries and ensure they are written efficiently. Use appropriate join types avoid unnecessary select *, and limit the result set with where cluase.
   c. If you are using hibernate or JPA for database access, understand the generated SQL queries and use features like lazy loading, batch fatching.
   d. by using pagination and fetch limit
2. **Microservice application, or monolithic**
   a. Four Peelers Of microservice -
      i. Single Responsibility Principle - a single service should be developed for single business logic.
      ii. Loose Coupling - all microservices should be loosly coupled, a change in one microservice do not have heavily impact on other
      iii. Autonomy - every microservice is autonoumous and can be developed, deployed and scaled independently.
      iv. Resilience - if one service fails it does not bring dawn the entire system
3. **Complex feature or changes you have done or devoloped?**
   a. Complex feature that i have done in my project is bank invoice discounting logic,
4. **Why do we use packages in java projects**
   a. Packages provide a way to establish visibility and documentation boudaries within codebase. By grouping related classes and interfaces into packges, developers can convay the intended purpose and usage of components through packge level docunentation.
   b. Controll over class loading and classpath
   c. Packages help prevent naming conflicts between classes and interfaces with the same name but different packages.
5. **What heppen in java if two packages have the same class name**
   a. In java, if two packages have classes with the same name, it does't directly lead to a conflict or an error.
6. **Are you aware of the static key word in java**
   a. static keyword used to declare variables, methods and nested classes that belong to the class itself rather than to instnace of class.
7. **Can a static block throw an exception**
   a. Yes, a static block in java can throw exception
8. **Can we override static methods in java**
   a. No we can not overrride static method in java. Static method in java can not be overriden.
   b. This is because static methods are not are not associated with the instance of a class , but with the class itself.
   c. Therefore, when a subclass inherits a static method from its parent class, it canot modify the behavior of the static method in any way
9. **Can you expalin the execution flow when the class is loaded with the static and non static method.**
   a. The execution flow involves the initialization of static variables, excecution of static block(if any), and loading of static methods.
10. **Do you know about the generic classes in java.**
    a. Yes, i am familier with the generic class with java
    b. Generic classes are classes that can work with any data type. They are defined with one or more type parameters enclosed with angel brackets.
11. **Can we use generics with arrays**
    a. we can't create the array generics, alternatively its more common to use generic with collections like List<T> , which are type safe to avoid array related issue
12. **What happen when an exception is thrown in a static initialization block**
    a. every time we crates the object there are two steps, i. class loaded and ii. object are instantiated
    b. every time you load the class it will called the static block, and then and then it will created the object
    c. so if an exception is thrown in a static block, the class will not fully initialised. this means the static variable or static method that come after the point of failure will not be executed.
    d. if a exception occurs during static initialization of a class, it can be occurs in to a class load error
13. **Can we throw exception from finally block**
    a. Yes, We can throw the exception from finally block but it is generally not recommoded
14. **Do you know about custom exceptions and inbuild exceptions**
    a. Inbuild excepitions are
       i. RuntimeException
       ii. NullPointerException
       iii. ArrayOutOfBoundException
    b. Custom Exceptions are the exceptions which are created by the user by its own, the custom exception class must be extends the exceptions class
15. **Which are better custom exceptions or buildin exceptions**
    a. According to me Custom Exceptions are always better
16. **Can you tell any inbuild exception**
    a. Runtime Exceptions
    b. NullPointer Exceptions
    c. ArrayOutOfBoundExceptions
17. **What is method overloading**
    a. Method overloading in java is a feature that allows to class it can have multiple method with the same name, but the they can have different number of parameters.
    b. Method overloading is compile time polymorphism
18. **Can we determine method overloading at run time**
    a. No, method overloading is a concept that is resolved at compile time, not at run time.
19. **How does java resolve overloading at compile time**
    a. Java resolves overloading at compile time by matching the method calls argument to the method signatures declared in the class.
    b. it selects the most specific method based on the number and type of argument provided.
20. **Have you used final keyword in your project(about final keyword)**
    a. Final keyword can be used with variables, classes, and method
    b. Final variable means we can not chaged its value after declaration.
    c. Final class can be defined as inheritence of that class not possible.
    d. Final method can be defined as, method overriding is not possible when declare a method with final keyword.
21. **Can you tell me real time use case of final keyword in real time application**
    a. Yes, in my Infra, there are category ids for my each products
    b. So i declared the category ids of products using final keyword in a common package, so the in a entire repository the category id should not change.
22. Can we change final reference of final object
    a. Yes we can chage the state of the final object
23. **Whats a impact of declaring a method as final in a class and the class is inherited by subclass?**
    a. if declared the method as final in class, measn that method can not be overridden by any subclasses.
24. **Do you know about inner class in java**
    a. Inner class is defined as class inside class
25. Do you know about java memory model
26. What is garbage collection in java
27. How would you investigate and fix memory leak in java, whats the step would you take
28. Do you know about design pattern
29. Have you created singleton before
30. How can we create singleton
31. **What is comparable interface in java?**

a. In the java the comparable is the interface is used to define the natural ordering of objects. It requires implementing the compareTo method, which compares the current object with another of the same type. This method returns a negagive integer, or a positive integer if the current object is less than equal to or grater than specified object.

32. **If you have class with the properties name, age, and salary if wants to sort the objects of the class on the basis of name how will you do it.?**
    a. Fot this i will imlements the comparable interface in the emloyee class, i will overide the compareTo method to compare the name property of the current object with the name of the other object. This way when i use collection.sort(), the object will be sort alphabetically by their names
33. **what is the difference between callable and runnable in java?**
    a. In java, both runnable and callable are interfaces used to represent tasks that can be executed by a thread,
    b. Runnable does not returns a result, callable interface can return a result
    c. Runnable typically used when you dont need to return a result from the task or handle checked exceptions.
    d. in Runnable we can not throw the exceptions, in callable we can throw the exceptions
    e. runnable have return type void, callable have generic return type
34. How you are managing excetions in our project?
35. Can you explain what is deep clonning? — neet to study
36. what is java 8 features you are aware of?
37. can you explain what is consumer? — need to study
38. **what are the different states of a thread in java?**
    a. new - the thread is created but not yet started. It is in this state the instance of a thread is created but the start method is not been called.
    b. runnable - the thread is ready to run and waiting for cpu to execute it. this state is combination of ready and running state
    c. blocked - The thread is waiting to aquire a lock or monitor to enter a synchronised block or method.
    d. waiting - the thread is waiting for another thread to perform a perticular action.
    e. timed waiting - the thread is wating for specific amount of time
    f. terminated - the thread has completed its execution. It either copleted the run method or terminated due to an unhanddled error.
39. **How can you create thread in java?**
    a. There are two method in java to create the thread,
        i. the first one is by extending thread class in class, (class MyClass extends Thread), and need to create run() method in our class, and when we create the object of class need to call the start method, in this way we can create the thread.
        ii. The second one is by imlementing the runnable interface
40. **What is a dead lock and how we can avoid it?**
    a. dead lock is the situation when the two or more threads are blocked forerver, each waiting for the other to release a resource they need to contineu execution.
    b. It can be avoided by follwing best practices like avoiding nestead locks, maintaining consistent lock ordering, using tryLock, applying timeouts
41. **what is solid principles in java?**
    a. The solid principles in java are five key design principles for writing clean and maintainable code:
        i. Single Responsibility Principle: A class shold have only one jon or responsiblity,
        ii. Open/closed Principle: Class should be open for extension but closed for modification. you should be able to add new functionality without changing the existing code.
        iii. Liskov substitution Principle:
        iv. Interface Segregation Principle
        v. Dependency Inversion Principle: High level module should not depend on low level modules. Both should depend on abstraction.
42. what are the collections that you have used?
43. **How do you decide that you have to use set or List?**
    a. You would use a list when you need an ordered collection that can contain duplicates.
    b. Use a set when you need a collection of unique elements where order doest matter.
44. what is que in collections?
45. **Explain the concept of object oriented programming and its priciple**
    a. Object oriented programming is a programming centered around the concept of objects which are instances of class.
    b. The key priciples of OOPs are
        i. Encapsulation - Encapsulation involes the building data and methods that operates on the data into a single unit called class,
        ii. Inheritence -
        iii. polymorphism - measn many forms and allows objects of diferent classes to be treated as objects of common superclass. It enables single function.
        iv. Abstraction - abstraction is process of hiding the complex information details an object and exposing only the necessary features to the user.
46. **Can you explain the exception handlling in java? what is the difference between checked and unchecked exceptins**
    a. Exception handlling in java is a way to handle the errors which comes at runtime or compile time, in java exception handlling can be done throw try catch, throw blocks to manage errors and prevent program crashes.
    b. checked exceptions - checked exceptions that must be either caught or declared in the method signature using 'throws' . THey are checked at compile time (e.g IO exceptions)
    c. Unchecke exceptions are not required to to be caught or decalred they are checked at runtime.(null pointer exceptions)
47. Thread Priority -
    a. we can set the thread Priority with the help of method setPriority()
    b. we can get the priority of the thread with the help of method getPriority()
    c. 1 is the lowest priority and 10 is the highest priority, and 5 is the default priority

Important Notes (Java)

Every class in java extends object class
Types of Error ->. thread Error, Io Error,
Types of Exception -> Runtime Exception, I/O Exception, SQL exception
Runtime Exception -> Arithmatic Exception, ArrayIndexoutof Bound Exception, null pointerException,
Runtime Exception also called checked Exceptions and I/O exceptions, SQL exceptions also called unchecked exceptions

Java Collections Quetions-

1. **Explain the collection Hierarchy**
    a. It start with the collection interface -, Collection is the root of the Java collection framework and most of the collections in java are inherited from this inter
2. The Three basic interfaces that implements the collection framework are
    a. List -
        i. List Contains The orderd element
        ii. List May Includes Duplicates
        iii. Supports Index based searches
        iv. List interface is being extended by the few classes
            1. ArrayList - Dynamic Resizing
                a. It is Dynamic Resizing
                b. Whenever it gets Full, it will automatically increased its size to 50% of its original size
                c. it is non-synchronous
            2. Linked List-
                a. Maintains The Insersion Order
                b. it is non synchronised
                c. does not support accessing elements randomly.
            3. Vector-
                a. Vector is synchronised.

        b. Maintains the insertion order
        c. vector increases its size by doubling the array size
    b. Queue
        i. Queue Follows the FIFO approachch
        ii. element add at rear end and remove from the front end
        iii. There is only one implemeting class for queue interface
            1. Priority Que
    c. Set
        i. Set Doesnot define an order for the elements hence index based search is not supported
        ii. Does Not Contains Duplicates
        iii. The set interface is being extended by the few classes
            1. Hash set
                a. Contains Only Unique Elements
                b. Only Null Elements Can be added
            2. Linked List Hashset
    d. Map - map does not implements the collections interface
        i. it represents the key value pair
        ii. it can only contains a unique key, can have duplicate values
        iii. It is uniq interface in the collection framework
        iv. Two classes which interface this map interface
            1. Hash Map
            2. Hash Table
3. **Why map doesn't extend the Collection Interface**
    a. The map interface in java follows the key value pair structure whereas the collection interface is a collection of objects which are stored in a structure manner with a specified access mechanism.
    b. The main reason map doesnot extend the collection interface is that the add method of the collection interface does not support the key-value pair like map interface.
4. **Difference Between fail-fast and fail-safe Iterators**
    a. Fail-Fast iterators throws the exception, when the one thred is iterating over the collection object and other thread modify collection, they are called fail-fast
    b. The fail-safe iterator does not throw any exception
5. **What do you understand by BlockingQueue?**
    a. BlockingQue represents a queue which is thread safe to put elements into, and take elements out of from. In other words, multiple thread can be inserting and taking elements concurrently from java Booking queue, without any concurrency issue.
6. **What is difference between synchronized collection and concurrent collection**
    a. Both Synchronised and concurrent collection classes provide thread-safety.
    b. The difference between them comes in performence, Scalability and how they achieve thread-safety.
    c. synchronised collections are slower because lockin
7. **Internal Working Of Hash Map-**
    a. Hashmap in java works on hashing principle where hash functions are used to link key and value in hashmap, Objects contains bot key and value are stroed using put method of hashmap and retrive by calling put method.

Spring boot section starts

Interview Quetions

1. **Advantages Using springboot over traditional spring**
    a. Simplified Configuration
        i. Spring Boot reduces the need for boilerplate configuration.
        ii. It provides sensible defaults for many configuration, reducing the amount of XML or java configuration you need to write
    b. Embedded Servers
        i. SpringBoot comes with embedded servers like tomcat, Jetty, reducing the deployement complexity
    c. Autoconfiguration
        i. Spring boots autoconfiguration feature automatically configurs spring application based on dependencies present in the classpath. This reduces manual configuration efforts
    d. Starter Dependencies
        i. Spring boot provides starter dependencies that bundle commonly used libraries and configuration for specific use cases.
        ii. This siplifies the dependency management and reduces the chances of dependency conflicts.
    e. Actuator
    f. Spring Security
    g. Externalised Configuration
    h. Simplified Dependency Management
    i. Rapid Development Access

1. **@Transactional annotation - explain**
    a. This annotation mostly i have used with my Create, update or Delete API
    b. when you annotate method with @Transactional Annotation, spring ensures that the methods execution is wrapped in a Transaction
    c. If the methods completes successfully, the transaction is commited.
    d. If an exception occurs during the method execution, the transaction is roled back, ensuring data consultancy.
2. **What is default server in spring boot application**
    a. In a spring boot application if you dont specify any embedded server configuration, The default server that is used is Tomcat.
3. **What is autoconfiguration**
    a. Autoconfiguration is feature that aim to simplyfy the setup and configuration of spring boot application
    b. It autometically configure the springboot application based on the dependencies present in the classpath and sensible defaults
    c. This approach reduces the amount of boilerplate code developers need to write, making it easier and faster to get a spring boot appliacation up and and running
4. **How does the spring boot decides what to configure automatically internally**
    a. SpringBoot Desides what to configure automatically internally through a combination of classpath scanning, condition annotation, and predefined configuration classes.
5. **Can you give the example of situation where we need to overide the default autoconfiguration.**
    a. Customising The Datasource Configuration
        i. Spring Boot Automatically configures a datasource, when it detects a database driver on the class path. Howwhever you might need to connect multiple datasource according to conditions.
6. **Can you tell me the steps to integrate the database in a springboot application from scratch**
    a. Choose your Database
    b. Setup Spring boot project
    c. Configure Database Connection -
        i. 0pen your application.properties or application.yml file add your database configuration in the that file

    d. Create Entity Classes
        i. Entity classes are those which represent your database tables,
    e. Create Repository Interface
        i. These Interfaces will provide CRUD operation for your entities
    f. Create Controller and service Layer and test your application
7. **How would you secure a springboot application**
    a. Dependancy Configuration -
        i. Ensure that your Spring Boot Application includes necessary dependancies for security.
        ii. The most common dependancy is 'spring-boot-starter-security'
    b. Configuration
    c. Authentication
    d. Authorization
    e. User Management
        i. Manage your accounts, roles and permissions within your application. You can store your details in a database or use an external authentication provider such as LDAP or Oauth.
    f. Session Management
    g. Password Management
8. **What is spring security**
    a. Spring Security is a powerful and customizable authentication and authorization framework for securing spring application.
    b. It provides comprehensive security features such as authentication, authorization, sassion management, to protect application from unauthorized access and security therets.
9. What is diffence between authentication and authorization
10. **Method level security in class**
    a. Method level security in class typically refers to contolling access to individual methods within a class based on the permissions or roles of the caller.
    b. Private Method, Public Method, protected method.
    c. You can annotate methods with security expression or annotation to specify who can invoke them. for example you can use @Preauthorise, @PostAuthorise, or @Secured annotations to define access control rules at the method level
11. **What is the role of embedded server in spring boot application**
    a. The embedded server in a Spring Boot application serves as the runtime environment for hosting and managing the application.
    b. It eliminates the need for deploying the application to a separate server, as its embedded within a application itself. This simplifies deployment and configuration, making it easier to develop and run spring boot application.
12. **Can you deploy a spring boot application as a traditional war file to an external server**
    a. Yes, we can deploy a spring boot application as a traditional WAR(Web Application Archieve) to an external server, such as Apache TomCat or any other server container that supports WAR deployement.
13. How springboot handles application configuration
14. **Whats the order of precidence in the configuration process**
    a. Command Line Arguments
    b. Java System Properties
    c. Environment variables
    d. External Configuration
    e. Profile-specific properties
15. If i give you a prebuilt project and if i ask you to optimise this project so what steps would you take
16. **Do you know about spring boot actuator**
    a. SpringBoot Actuator is a production ready feature with the help of we can manage and monitor application
    b. After Adding spring boot actuator dependency in our application, it expose the one /actuator endpoit bydefault
    c. with the help of endpoints we can monitor and manage our spingboot application
17. **Can you name some endpoints provide by spring boot actuators**
    a. /beans, /actuator, /catches, /conditions
18. **Can we customies springboot actuators**
    a. Yes, we can customiser sprinboot actuators
19. Do you know some annotations in springboot before writing junit test cases
20. Do you know about the spring boot starters

<u>Top 12 Interview Quetions For Spring Boot</u>

1. **Is it possible to change the port of the embedded tomcat server in spring boot**
    a. Yes, we can chage the port by mentioning server.port in application.properties or in application.yml file
2. **Can we override or replace the embedded tomcat server in spring boot**
    a. Yes we change replace the embedded tomcat server with any server by using the starter dependency in the pom.xml file
3. **What is auto configuration in spring boot ? How does it help ? why spring boot is called opineonated?**
    a. Autoconfiguration is feature that aim to simplyfy the setup and configuration of spring boot application
    b. It autometically configure the springboot application based on the dependencies present in the classpath and sensible defaults
4. **What is the difference between @SpringBootAplication and @EnableAutoConfigurationAnnotation?**
    a. @EnableAutoConfiguration is the annotation to ennable the Autoconfiguration
    b. @SpringBootAplication = @configuration + @scanclasspath + @EnableAutoConfigurationAnnotation
5. **What is the difference between @Restcontroller and @Controller in spring boot?**
    a. Both @controller and @Restcontrolle are used to define the webcontrollers,
    b. @RestController is the combination @ResponseBody + @Controller ,
    c. @ResponseBody is annotation in spring boot is used to indicate that the return value of method should be written directly to the http response_body
    d. @Controller - Map of the model object to view or template and make it human redable
    e. Simply returns the object and reject the object data is directly written in HTTP response as JSON or XML
    f. @RestController = @Controller + @ResponseBody
6. **Describe the flow of HTTPS requests through the spring boot application**
    a. client -> https req -> controller layer -> service-layer
7. what is the difference between @RequestMapping and @GetMapping?
    a. RequestMapping -> can be used with GET PUT POST and many other request methods using the method attribute on the annotation
        i. E.g @RequestMapping(value = "/user/{userId}", method = RequestMethod.GET
    b. @GetMapping is only extension of @RequestMapping with get method which helps you to improve on clarity on request
8. What is the difference between the embedded container and the WAR
9. **What is the use of Profiles in the springboot**
    a. Spring has the provision of profile to keep the seperate configuration of environment

<u>Other Quetions</u>

1. **What is spring boot**
    a. Spring boot is java framebwork that makes it easier to create and run java application
    b. It simplifies the configuration and setup process, allowing developers to focus more on writing code for their applications
    c. Spring boot is a module of spring framework, facilities Rapid Application Development(RAD) capabilities.
    d. Spring boot solve many developers problem,
        i. Configuration

   ii. Dependency Management
   iii. Embedded Server
2. **Why springboot over spring?**
   a. spring boot provided many advantages over sprig framework
   b. remove boilerplate code
   c. provide production ready application - features like metrix, helth check and many features are designed for production ready application
   d. Rapid Development - auto configuration enable developers to quickly develops app
3. **Working Of Springboot**
   a. spring boot starts by scanning the starter dependencies in pom.xml
   b. Then download and auto-configure the module as you included in pom.xml
   c. For example, if you want to create one web application, then for that we have to put spring-boot-starter-web dependency in pom.xml, then when we starts the project spring boot will dawnload all the dependency required for web and configure the things like spring mvc
4. **How spring boot starts**
   a. Starts by calling main() method of your main class
   b. The run() method of SpringApplication is called. This Method starts the application by creating an application context and initializing it.
   c. Once the application context is initialized, the run() method stats the applications embedded web server.
5. **SpringBoot Top Annotation**
6. **What are the springboot starters?**
   a. Starters are a collection of pre-configured dependencies that make it easier to develop perticular kinds of application
   b. These starters include all the dependencies , version control and configuration
7. **What are the key dependencies of Spring Boot**
   a. Springboot Starter parent
   b. Springboot maven plugin
   c. Springboot starter test
   d. Springboot starter security
   e. Springboot Starter actuator
   f. springboot starter web
8. **What is spring boot starter Parent?**
   a. Spring Boot Starter Parent is a starter project that provides the default configuration for spring-based applications
   b. The dependency management feature manages the versions of common dependencies.
   c. Provide the default compiler level as java 1.8 and UTF-source encoding.
   d. provides a default configuration for maven plugins such as, maven-surefire-plugin, maven-jar-plugin, and maven-failsafe-plugin
   e. Executes a repackages goal with a repackages execution id.
9. **Can we use Only SpringBoot Dependencies feature and configure maven plugin manually**
   a. Yes we can,
   b. For that, we have to remove SpringBoot-starter-parent from dependecies
   c. include the spring-boot-dependency dependency inside the dependency management section as an import scope.
10. **What is Spring boot CLI and what are its benefits ?**
    a. CLI is command line tool to create, run and manage spring boot application.
11. What is thymeLeaf

Spring Boot

7. **@SpringBootApplication** ->
   a. It is Core annotation of springboot
   b. This annotation is a combination of three annotation -> @Configuration, @EnableAutoConfiguration, @ComponentScan
8. **@Configuration -**>
   a. About this annotation, whatever the class which is annotated with this annotation, we can declare one or more beans in the class which is annotated with the @Configuration
   b. Bean is basically, if we have created one class and we dont want to maintain object of that class, like spring should handle the object of that class,
9. **@ComponentScan**
   a. This annotation is used to all packages in your spring boot application, to scan all configuration classes, all dependencies.
   b. If you have one more package in your application rather than your main package, then your @SpringBoot annotation will handle your main package, and if your other package contain configuraion class, and if that class is not annotated with @Component class then your configuration class will not be work
10. **@Bean** ->
    a. this annotation is used in class which is annotated with the @Configuration class, this is method level annotation
    b. whatever the methods which are annotated with the @Bean annotation, that method wil handle by spring container
11. **@Component ->**
    a. if You dont want to create a bean of class and also you want to handle the object of that class by spring container then annotate that class with @Component
12. **@Autowired**
    a. This annotation is used for dependency injection, if you want to use one class in other class, declare that class in other class with @Autowired annotation, and spring container will give you the object of that class.
13. **@Qualifire-**
    a. This annotation is used to avoid the conflict of dependency injection
14. **@Service**
    a. In an spring boot application, the business logic resides within the service layer so we use the @Sevice annotation to indicate that a class belongs to that layer.
    b. @Service annotation can be applied only to class
    c. it is used to mark the class as a service provider. So overall @Service annotation is used with classes that provide some business functionalities.
    d. Spring context will autodetect these classes when anotation-based configuration and classpath scanning is used

Spring Boot Notes

1. **@Controller**
   a. when you annotate class with @controller annotation, spring boot identifies it as controller component
   b. Controllers in spring boot are responsible for handlling incoming appropriate responses
2. **@RequestBody** - When a controller method is annotated with a @RequestBody, Spring boot application automatically converts the body of incoming http request to the method parameter type
   a. It uses message converter to deserialise the request body into the java object.
3. **@ResponseBoy** - @ResponseBody Annotation used to indicate that the return value of controller method should be directly written to the http response body