**Q) List down the new features introduced in Java 8?**

**Answer:** **New features that are introduced in Java 8 are enlisted below:**

* Lambda Expressions
* Method References
* Optional Class
* Functional Interface
* Default methods
* Nashorn, JavaScript Engine
* Stream API
* Date API

**Q) What are Functional Interfaces?**

**Answer:** Functional Interface is an interface that has only one abstract method. The implementation of these interfaces is provided using a Lambda Expression which means that to use the Lambda Expression, you need to create a new functional interface or you can use the predefined functional interface of java.

The annotation used for creating a new Functional Interface is “**@FunctionalInterface**”.

**Q) What is an optional class?**

**Answer:** Optional class is a special wrapper class introduced in Java 8 which is used to avoid NullPointerExceptions. This final class is present under java.util package. NullPointerExceptions occurs when we fail to perform the Null checks.

**Q) What are the default methods?**

**Answer:** Default methods are the methods of the Interface which has a body. These methods, as the name suggests, use the default keywords. The use of these default methods is “Backward Compatibility” which means if JDK modifies any Interface (without default method) then the classes which implement this Interface will break.

On the other hand, if you add the default method in an Interface then you will be able to provide the default implementation. This won’t affect the implementing classes.

**Syntax:**

|  |
| --- |
| public interface questions{            default void print() {    System.out.println("www.softwaretestinghelp.com");               }      } |

**Q) What was wrong with the old date and time?**

**Answer:** **Enlisted below are the drawbacks of the old date and time:**

* Java.util.Date is mutable and is not thread-safe whereas the new Java 8 Date and Time API are thread-safe.
* Java 8 Date and Time API meets the ISO standards whereas the old date and time were poorly designed.
* It has introduced several API classes for a date like LocalDate, LocalTime, LocalDateTime, etc.
* Talking about the performance between the two, Java 8 works faster than the old regime of date and time.

**Q) How can you create a Functional Interface?**

* **Answer:** Although Java can identify a Functional Interface, you can define one with the annotation
* **@FunctionalInterface**
* Once you have defined the functional interface, you can have only one abstract method. Since you have only one abstract method, you can write multiple static methods and default methods.

**What is Method Reference?**

* **Answer:** In Java 8, a new feature was introduced known as Method Reference. This is used to refer to the method of functional interface. It can be used to replace Lambda Expression while referring to a method.
* **For Example:** If the Lambda Expression looks like
* num -> System.out.println(num)
* Then the corresponding Method Reference would be,
* System.out::println
* where “::” is an operator that distinguishes class name from the method name.

**Q) What is a Stream API? Why do we require the Stream API?**

**Answer:** Stream API is a new feature added in Java 8. It is a special class that is used for processing objects from a source such as Collection.

**We require the Stream API because,**

* It supports aggregate operations which makes the processing simple.
* It supports Functional-Style programming.
* It does faster processing. Hence, it is apt for better performance.
* It allows parallel operations.

**Q) What is the difference between limit and skip?**

**Answer:** The limit() method is used to return the Stream of the specified size. **For Example,** If you have mentioned limit(5), then the number of output elements would be 5. Whereas, the skip() method is used to skip the element.

**Q)** **Write a Java 8 program to get the sum of all numbers present in a list?**

**package** com.yash.Java8;

**import** java.util.ArrayList;

**class** Java8 {

**public** **static** **void** main(String[] args) {

ArrayList<Integer> list = **new** ArrayList<Integer>();

list.add(10);

list.add(20);

list.add(30);

list.add(40);

list.add(50);

System.***out***.println(*sum*(list));

}

**public** **static** **int** sum(ArrayList<Integer> list) {

**return** list.stream().mapToInt(Integer::intValue).sum();

}

}

**Spring Boot Interview Question Answer**

### **Q) What is Spring Boot Actuator?**

Spring Boot provides actuator to monitor and manage our application. Actuator is a tool which has HTTP endpoints. when application is pushed to production, you can choose to manage and monitor your application using HTTP endpoints.

**Q) What are the advantages of Spring Boot?**

* Create stand-alone Spring applications that can be started using java -jar.
* Embed Tomcat, Jetty or Undertow directly. You don't need to deploy WAR files.
* It provides opinionated 'starter' POMs to simplify your Maven configuration.
* It automatically configure Spring whenever possible.

**Q) How to create Spring Boot application using Maven?**

There are multiple approaches to create Spring Boot project. We can use any of the following approach to create application.

* Spring Maven Project
* Spring Starter Project Wizard
* Spring Initializr
* Spring Boot CLI

**Q) How to create Spring Boot project using boot CLI?**

It is a tool which you can download from the official site of Spring Framework. Here, we are explaining steps.

### Q) What are the Spring Boot Annotations?

The @RestController is a stereotype annotation. It adds @Controller and @ResponseBody annotations to the class. We need to import org.springframework.web.bind.annotation package in our file, in order to implement it.

**Q) What is Spring Boot dependency management?**

Spring Boot manages dependencies and configuration automatically. You don't need to specify version for any of that dependencies.

Spring Boot upgrades all dependencies automatically when you upgrade Spring Boot.

### **Q) What are the Spring Boot Starters?**

Starters are a set of convenient dependency descriptors which we can include in our application.

Spring Boot provides built-in starters which makes development easier and rapid. For example, if we want to get started using Spring and JPA for database access, just include the **spring-boot-starter-data-jpa** dependency in your project.

### ****Q)** How to connect Spring Boot to the database using JPA?**

Spring Boot provides **spring-boot-starter-data-jpa** starter to connect Spring application with relational database efficiently. You can use it into project POM (Project Object Model) file.

### ****Q)** What is @RestController annotation in Spring Boot?**

The @RestController is a stereotype annotation. It adds @Controller and @ResponseBody annotations to the class. We need to import org.springframework.web.bind.annotation package in our file, in order to implement it.

### ****Q)** What is @RequestMapping annotation in Spring Boot?**

The **@RequestMapping** annotation is used to provide routing information. It tells to the Spring that any HTTP request should map to the corresponding method. We need to import org.springframework.web.annotation package in our file.

**Typescript Interview Question and answer**

### ****Q) List some benefits of using Typescript?****

TypeScript has the following benefits.

* It provides the benefits of optional static typing. Here, Typescript provides types that can be added to variables, functions, properties, etc.
* Typescript has the ability to compile down to a version of JavaScript that runs on all browsers.
* TypeScript always highlights errors at compilation time during the time of development whereas JavaScript points out errors at the runtime.
* It helps in code structuring.
* It uses class-based object-oriented programming.
* It provides excellent tooling supports with IntelliSense which provides active hints as the code is added.
* It has a namespace concept by defining a module.

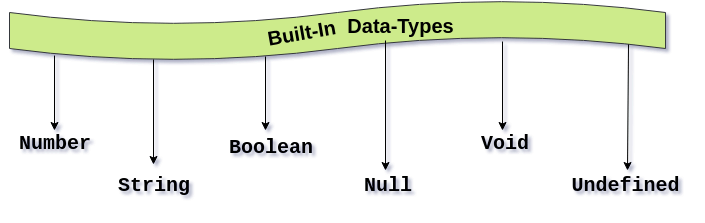
### **Q) What are the disadvantages of TypeScript?**

TypeScript has the following disadvantages:

* TypeScript takes a long time to compile the code.
* TypeScript does not support abstract classes.
* To use any third party library, the definition file is must. And not all the third party library have definition file available.
* Quality of type definition files is a concern as for how can you be sure the definitions are correct?

### ****Q) List the built-in types in Typescript.****

The built-in data types are also known as primitive data types in Typescript. These are given below.



### **Q) What are the variables in Typescript? How to create a variable in Typescript?**

We can declare a variable in one of the four ways:

1. Declare type and value in a single statement. Syntax: var [identifier] : [type-annotation] = value;
2. Declare type without value. Syntax: var [identifier] : [type-annotation];
3. Declare its value without type. Syntax: var [identifier] = value;
4. Declare without value and type. Syntax: var [identifier];

**Q) What is tsconfig.json file?**

The tsconfig.json file is a file which is in JSON format. In the tsconfig.json file, we can specify various options to tell the compiler how to compile the current project. The presence of a tsconfig.json file in a directory indicates that the directory is the root of a TypeScript project.

**Q) Explain generics in TypeScript?**

TypeScript Generics is a tool which provides a way to create reusable components. It is able to create components that can work with a variety of data types rather than a single data type. Generics provides type safety without compromising the performance, or productivity. Generics allow us to create generic classes, generic functions, generic methods, and generic interfaces.

**Q) Does TypeScript support all object-oriented principles?**

Yes, TypeScript support all object-oriented principles. There are four main principles to object-oriented programming:

1. Encapsulation,
2. Inheritance,
3. Abstraction, and
4. Polymorphism.

**Angular interview question and answer**

### ****Q) What is Angular? / What do you know about Angular?****

Angular is one of the most popular JavaScript frameworks developed and maintained by Google. It is an open-source front-end web framework based on TypeScript. It is most suited for developing enterprise web applications because its code is reusable and maintainable.

### ****Q) What is the main purpose of Angular?****

The main purpose of using Angular is to create fast, dynamic and scalable web applications. We can create these applications very easily with Angular using components and directives.

### **Q) What are the biggest advantages of using Angular?**

Following is the list of the biggest advantages of using the Angular framework:

* Angular supports two-way data-binding.
* It follows MVC pattern architecture.
* It supports static templates and Angular template.
* It facilitates you to add a custom directive.
* It also supports RESTfull services.
* Validations are supported in Angular.
* Angular provides client and server communication.
* It provides support for dependency injection.
* It provides powerful features like Event Handlers, Animation, etc.

### **Q) What are templates in Angular?**

In Angular, templates contain Angular-specific elements and attributes. These are written with HTML and combined with information coming from the model and controller, which are further rendered to provide the user's dynamic view.

### ****Q) What are directives in Angular?****

A directive is a class in Angular that is declared with a @Directive decorator. Every directive has its own behavior, and you can import them into various components of an application.

### **Q) What is Angular CLI?**

Angular CLI is a short form for Angular Command Line Interface. It is a command-line interface to scaffold and build angular apps using node.js style modules.

To use Angular CLI, we have to install it by using the following **npm** command:

1. npm install @angular/cli@latest