Question 1: Which of the following component intercepts all requests in a Spring MVC application?

FilterDispatcher

**DispatcherServlet**

ControllerServlet

Question 2: Which annotation is used to map a request to a method of a controller?

**RequestMapping**

RequestMapper

URLMapper

RequestResolver

Question 3: Which of the following can not be the return type of a request processing method of a controller?

String

Void

**Object**

ModelAndView

Question 4: Which annotation is required to receive a model object initialized by the request parameters, in a request processing method?

@RequestParam

**@ModelAttribute**

@ModelObject

Question 5: --- is required to load the beans configured in ServletInitializer class?

ContextLoader

ContextLoaderListener

ApplicationContextLoader

**WebApplicationInitializer**

Question 6: What should be the return type of a request processing method of a controller to process an REST request?

**String**

ModelAndView

Void

View

Question 7: To apply custom validation, which of the following interface need to be implemented?

**Validator**

ValidationAware

Validatable

Question 8: --- object is used by a request processing method to check validation failure?

ValidationErrors

**BindingResult**

ValidationBinding

Question 9: Which of the following class need to be extended in the java beans configuration file for configuring static resource handling?

mvc

**WebMvcConfigurerAdapter**

aop

Question 10: Which of the following ViewResolver is required for Tiles?

InternalResourceViewResolver

**UrlBasedViewResolver**

BeanNameViewResolver

Question 11: Spring Web MVC (and in Spring in general) is the "Open for extension, closed for modification" principle.The reason that this principle is being mentioned here is because a number of methods in the core classes in Spring Web MVC are marked final.

**true**

false

Question 12: WebApplicationContext has more scopes like request, session, globalSession to the standard Application context (which only provides Singleton and Prototype scopes)

**true**

false

Question 13: Which statements are true about DispatcherServlet ?

DispatcherServlet is an expression of the "Front Controller" design pattern .

The DispatcherServlet is the actual Servlet .

DispatcherServlet is initialized in the ServletInitializer class of your web application.

The Spring DispatcherServlet has a couple of special beans it uses in order to be able to process requests and render the appropriate views. These beans can be configured in WebApplicationContext.

**All the above**

Question 14: Which statements are true about WebApplicationContext ?

The WebApplicationContext is an extension of the plain ApplicationContext

The WebApplicationContext is bound in the ServletContext

**WebApplicationContext differs from a normal ApplicationContext in that it has all necessary Spring MVC components**

We can get WebApplicationContext reference by using ApplicationContext .

All the above

Question 15: @RequestParam is useful for binding query parameters to method parameters .

**true**

false

Question 16 - Which of the following class can be used to execute Sql queries in spring?

JDBCHelper

**JdbcTemplate**

DBHelper

DBTemplate

Question 17 What is Dependency Injection?

**A - It is a design pattern which implements Inversion of Control for software applications.**

B - It is one of the spring module.

C - It is a technique to get dependencies of any project.

D - It is used to promote tight coupling in code.

Question 18 If a bean can be created any number of times, scope is

A - session

B - global-session

**C - prototype**

D – request

Question 19 - Which is the way to provide configuration metadata to spring?

A - XML Based configuration file.

B - Annotation based configuration.

C - Java based configuration.

**D - All of the above.**

Question 20 - What is byType mode of autowiring?

A - Default setting which meas no autowiring and you should use explicit bean reference for wiring.

B - Autowiring by property name. Spring tries to match and wire its properties with the beans defined by the same names in the configuration file.

C - Spring first tries to wire using autowire by constructor, if it does not work, Spring tries to autowire by byType.

**D - Autowiring by property type. Spring tries to match and wire a property if its type matches with exactly one of the beans name in configuration file.**