1. What is codeigniter?

Codeigniter is open source web application framework.Its is for building websites using php.Codeigniter is loosely based on MVC pattern.Most simple framework in php , which is you will easily learn.Its mostly known for its speed as compare to other frameworks in php. Its goal is to enable you to develop projects much faster than you could if you were writing code from scratch, by providing a rich set of libraries for commonly needed tasks, as well as a simple interface and logical structure to access these libraries.

**2. When and who developed codeigniter?**

The first public version of CodeIgniter was released on February 28, 2006.

**3. What is current version of codeigniter?**

The latest is **Version 2.1.4**, till Sep 2013.

**4. What are the features of codeigniter?**

1. Codeigniter is free to use,its an open source framework.

2. Its light weight.The core system requires only a few very small libraries.Not like other frameworks that require heavy file libraries.

3. CodeIgniter is Fast.Its faster than any other framework in php.

4. The URLs generated by CodeIgniter are clean and search-engine friendly.Youwill change any url to what ever you want from files.

5. CodeIgniter is Extensible.The system can be easily extended through the use of your own libraries, helpers, or through class extensions or system hooks.

6. CodeIgniter Uses MVC(Model View Controller) which allows great separation between logic and presentation.

7. CodeIgniter requires nearly zero configuration,does not require you to use the command line,not forced to learn a templating language.

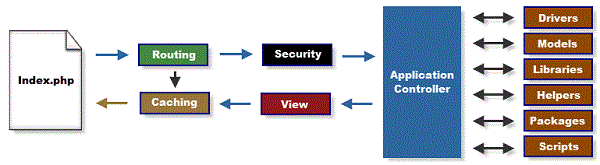
8. Full Featured database classes with support for several platforms,Security and XSS Filtering,Error Logging.

**5. Explain Codeigniter File Structure.**

When you download Codeigniter you will see the following folder structure :-

* **application**
  + cache
  + Config
  + Controllers
  + core
  + errors
  + helpers
  + hooks
  + language
  + libraries
  + logs
  + models
  + thirdparty
  + views
* **system**
  + core
  + database
  + fonts
  + helpers
  + language
  + libraries

**6. Explain Application Flow Chart in codeigniter.**

[](http://websexpert.net/uncategorized/codeigniter-interview-questions-and-answers.html/attachment/appflowchartcodeigniter)

Application flow chart from Codeigniter documentaion

1. The index.php serves as the front controller, initializing the base resources needed to run CodeIgniter.  
2. The Router examines the HTTP request to determine what should be done with it.

3. If a cache file exists, it is sent directly to the browser, bypassing the normal system execution.

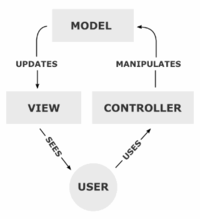
4. Security. Before the application controller is loaded, the HTTP request and any user submitted data is filtered for security.

5. The Controller loads the model, core libraries, helpers, and any other resources needed to process the specific request.

6. The finalized View is rendered then sent to the web browser to be seen. If caching is enabled, the view is cached first so that on subsequent requests it can be served.

**7. Explain MVC in Codeigniter.**

Model–View–Controller (MVC) is an architecture that separates the representation of information from the user’s interaction with it.

[](http://websexpert.net/php5/what-is-mvcmodel-view-controller-architecture.html/attachment/mvc-process)

**Controller:-** The **Controller** serves as an *intermediary* between the Model, the View. *controller* mediates input, converting it to commands for the model or view.

**Model:-**The **Model** represents your data structures. Typically your model classes will contain functions that help you retrieve, insert, and update information in your database.The *model* consists of application data and business rules.

**View:-**The **View** is the information that is being presented to a user. A View will normally be a web page.A *view* can be any output representation of data.

For more detail understanding MVC please read this article [What is MVC(Model-View-Controller) Architecture.](http://websexpert.net/php5/what-is-mvcmodel-view-controller-architecture.html)

**8. What are the hooks in codeigniter?**

CodeIgniter’s Hooks feature provides modify the inner workings of the framework without hacking the core files.How ever you like to cause some action to take place at a particular stage in the execution process.

the hooks feature can be globally enabled/disabled by setting the following item in the application/config/config.php file:

$config['enable\_hooks'] = TRUE;

Hooks are defined in application/config/hooks.php file.For example

|  |  |
| --- | --- |
| 1 | $hook['pre\_controller'] = array( |
| 2 | 'class'    =&gt; 'MyClass', |

|  |  |
| --- | --- |
| 3 | 'function' =&gt; 'Myfunction', |
| 4 | 'filename' =&gt; 'Myclass.php', |

|  |  |
| --- | --- |
| 5 | 'filepath' =&gt; 'hooks', |
| 6 | 'params'   =&gt; array('test', 'test1', 'webs') |

|  |  |
| --- | --- |
| 7 | ); |

**9. How you will add or load an model in codeigniter?**

Models will typically be loaded and called from within your controller functions. To load a model you will use the following function:

|  |  |
| --- | --- |
| 1 | $this->load->model('Model\_name'); |

**10. What are the helpers in codeigniter?**

Each helper file is simply a collection of functions in a particular category.There are URL Helpers, that assist in creating links, there are Form Helpers that help you create form elements, Text Helpers perform various text formatting routines, Cookie Helpers set and read cookies, File Helpers help you deal with files, etc.

Loading a helper file is quite simple using the following function:

|  |  |
| --- | --- |
| 1 | $this->load->helper('name'); |

**11. How you will use or add codeigniter libraries?**

All of the available libraries are located in your system/libraries folder. In most cases, to use one of these classes involves initializing it within a controller using the following initialization function:-

|  |  |
| --- | --- |
| 1 | $this->load->library(‘name'); |

**12. How you will work with error handling in codeigniter?**

CodeIgniter lets you build error reporting into your applications using the functions:-

**1. show\_error():-** This function will display the error message supplied to it using template application/errors/error\_general.php.

**2. show\_404() :-** Function will display the 404 error message.

**3.** **log\_message(‘level’, ‘message’) :-** This function lets you write messages to your log files. You must supply one of three “levels” in the first parameter, indicating what type of message it is (debug, error, info), with the message itself in the second parameter.

1.What is codeigniter ?

Codeigniter is a framework which is mainly focussed on web application development .It is well known for its performance speed.It has inbuilt libraries which helps to build web application faster.

2.Who developed codeigniter?

Codeigniter was developed bt ellislab Inc.

3.Why codeigniter is called as loosely based mvc framework?

Reason behind this is we does not need to follow strict mvc pattern while creating application.We can also able to build with model only view and controllers are enough to built a application.

4.What are helpers?

It is a group of functions which helps you to do particular functions. Example there are form helper to create a form .$this->load->helper(‘form’);

5.What are hooks in codeigniter?

Codeigniter’s hooks feature provides a way to change the inner working of the framework without hacking the core files. In other word, hooks allow you to execute a script with a particular path within the Codeigniter.  Usually, it is defined in application/config/hooks.php file.

**6) Explain routing in Codeigniter?**

In CodeIgniter, the way PHP files served is different rather than accessing it directly from the browser. This process is called routing. Routing in CodeIgniter gives you freedom to customize the default URL pattern to use our own URL pattern according to the requirement. So, whenever there is a request made and matches our URL pattern it will automatically direct to the specified controller and  function.

**7) Why is there a need to configure the URL routes?**

Changing the URL routes has some benefits like

* From SEO point of view, to make URL SEO friendly and get more user visits
* Hide some URL element such as a function name, controller name, etc. from the users for security reasons
* Provide different functionality to particular parts of a system

**8) List out different types of hook point in Codeigniter?**

Different types of hook point in Codeigniter includes

* post\_controller\_constructor
* pre\_controller
* post\_sytem
* pre\_system
* cache\_override
* display\_override
* post\_controller

**9) Mention what are the security parameter for XSS in CodeIgniter?**

Codeigniter has got a cross-site scripting hack prevention filter. This filter either runs automatically or you can run it as per item basis, to filter all POST and COOKIE data that come across.  The XSS filter will target the commonly used methods to trigger JavaScript or other types of code that attempt to hijack cookies or other malicious activity. If it detects any suspicious thing or anything disallowed is encountered, it will convert the data to character entities.

**10) Explain how you can link images/CSS/JavaScript from a view in code igniter?**

In HTML, there is no Codeigniter way, as such it is a PHP server side framework. Just use an absolute path to your resources to link images/CSS/JavaScript from a view in CodeIgniter

/css/styles.css

/js/query.php

/img/news/566.gpg

**11) Explain what is inhibitor in CodeIgniter?**

For CodeIgniter, inhibitor is an error handler class, using the native PHP functions like set\_exception\_handler, set\_error\_handler, register\_shutdown\_function to handle parse errors, exceptions, and fatal errors.

**12) Mention what is the default URL pattern used in Codeigniter framework?**

Codeigniter framework URL has four main components in default URL pattern.  First we have the server name and next we have the controller class name followed by controller function name and function parameters at the end. Codeigniter can be accessed using the URL helper. For example**http://servername/controllerName/controllerFunction/parameter1/parameter2.**

**13) Explain how you can prevent CodeIgniter from CSRF?**

There are several ways to protect CodeIgniter from CSRF, one way of doing is to use a hidden field in each form on the website.  This hidden field is referred as CSRF token; it is nothing but a random value that alters with each HTTP request sent. As soon as it is inserted in the website forms, it gets saved in the user’s session as well.  So, when the form is submitted by the users, the website checks whether it is the same as the one saved in the session. If it is same then, the request is legitimate.

**14) Explain how you can enable CSRF (Cross Site Request Forgery) in CodeIgniter?**

You can activate CSRF (Cross Site Request Forgery) protection in CodeIgniter by operating your application/config/config.php file and setting it to

$config [ ‘csrf\_protection’] = TRUE;

If you avail the form helper, the **form\_open()** function will insert a hidden csrf field in your forms automatically

**15) Explain how you can extend the class in Codeigniter?**

To extend the native input class in CodeIgniter, you have to build a file named application/core/MY\_Input.php and declare your class with

Class MY\_Input extends CI\_Input {

}

***16). How you will work with error handling in codeigniter?***

*CodeIgniter lets you build error reporting into your applications using the functions described below. In addition, it has an error logging class that permits error and debugging messages to be saved as text files.*

***show*error('message' [, int $status*code= 500 ] )***

*This function will display the error message supplied to it using template application/errors/error*general.php.

**show*404('page' [, 'log*error'])**

This function will display the 404 error message supplied to it using template application/errors/error*404.php.*

***log*message('level', 'message')**

This function lets you write messages to your log files. You must supply one of three "levels" in the first parameter, indicating what type of message it is (debug, error, info), with the message itself in the second parameter.

Indexing means it will create separate ordering memory for that particular filed for fast searching . if we assign index to the particular field means we can search specific row with that field quickly

3) How can you initilize arguments while creating function.

A) function test($a=false)

4) $config['encryption\_key'] = 'howAreYou'

It's important for you to know that the encoded messages the encryption function generates will be approximately 2.6 times longer than the original message. For example, if you encrypt the string "my super secret data", which is 21 characters in length, you'll end up with an encoded string that is roughly 55 characters (we say "roughly" because the encoded string length increments in 64 bit clusters, so it's not exactly linear). Keep this information in mind when selecting your data storage mechanism. Cookies, for example, can only hold 4K of information.