



This file will be divided into 4 sections as follows:

- 1. Contain 6 Concepts questions (understand them without playing on your mind).**
- 2. Contain coding with its output..**
- 3. Contain shorts coding in my point of view .**
- 4. Contain extra question.**
- 5. Contain long coding(check out the last revision and I hope that the long answer may contain the (XML assignment) and php revision array.**

Sections 1

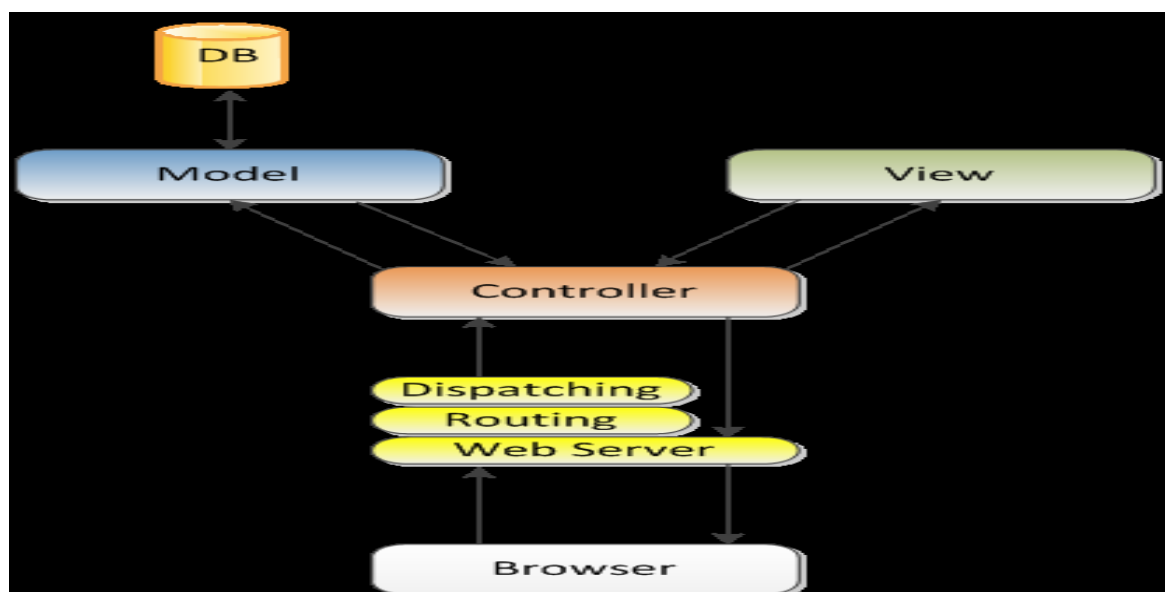
Contain 6 Concepts questions (understand them without playing on your mind).

1-Describe the three separate responsibilities in the MVC pattern and how each interacts with the other.

ANSWER:

The controller is responsible for receiving the request and validating the input, the model interacts with the database, and the view produces human-consumable renderings of the result of the request. The sequence of interaction is that:

- a. The controller receives the request and validates the inputs
- b. The controller invokes the model to query or update the database.
- c. The model returns the requested data.
- d. The controller transforms the data into something the view can use and forwards the data to the view.
- e. The view renders the result in a human-readable format and returns it to the controller.
- f. The controller returns the rendered HTML to the web server.



2. What are the seven steps needed to use a database from within Java?

ANSWER:

- Load the JDBC driver
- Define the connection URL
- Establish the connection
- Create a Statement object
- Execute a query or update
- Process the results
- Close the connection.

3. What is the using of database transactions explain ?

When a database is updated, by default the changes are permanently written (or committed) to the database. However, this default behavior can be programmatically turned off. If auto committing is turned off and a problem occurs with the updates, then each change to the database can be backed out (or rolled back to the original values). If the updates execute successfully, then the changes can later be permanently committed to the database. This approach is known as **transaction management**.

4. What are Design Patterns explain them with your own words ?

Strategy Pattern

The **strategy pattern** is typically used when your programmer's algorithm should be interchangeable with different variations of the algorithm. For example, if you have code that creates an image, under certain circumstances, you might want to create JPEGs and under other circumstances, you might want to create GIF files

Singleton Pattern

The **singleton pattern** is probably one of the best-known design patterns. You have probably encountered many situations where you have an object that handles some centralized operation in your application, such as a logger object. In such cases, it is usually preferred for only one such application-wide instance to exist and for all application code to have the ability to access it.

Factory Pattern

Polymorphism and the use of base class is really the center of OOP. However, at some stage, a concrete instance of the base class's subclasses must be created. This is usually done using the factory pattern

Observer Pattern

PHP applications, usually manipulate data. In many cases, changes to one piece of data can affect many different parts of your application's code. For example, the price of each product item displayed on an e-commerce site in the customer's local currency is affected by the current exchange rate. Now, assume that each product item is represented by a PHP object that most likely originates from a database; the exchange rate itself is most probably being taken from a different source and is not part of the item's database entry

5. What are the techniques that make the script “SAFE”?

Input Validation

For different kinds of input, you can use different methods. For instance, if you expect a parameter passed with the HTTP GET method to be an integer, force it to be an integer in your script.

HMAC Verification

If you need to prevent bad guys from tampering with variables passed in the URL (such as for a redirect as shown previously, or for links that pass special parameters to the linked script), you can use a hash, as shown in the following script.

Input Filter

By using PHP 5, you can add hooks to process incoming data, but it's mainly targeted at advanced developers with a sound knowledge of C and some knowledge of PHP internals. These hooks are called by the SAPI layer that treats the registering of the incoming data into PHP. One appliance might be to `strip_tags()` all incoming data automatically. Although all this can be done in user land with a function such as `sanitize_vars()`, this solution can only be enforced by writing a script that performs the desired processing and setting `auto_prepend_file` in `php.ini` to designate this script. Setting `auto_prepend` causes the processing script to be run at the beginning of every script. On the other hand, the server administrator can enforce a solution. For information on this, see http://www.derickrethans.nl/sqlite_filter.php for an implementation of a filter that uses SQLite as an information source for filter rules.

Working with Passwords

Another appliance of hash functions is authenticating a password entered in a form on your web site with a password stored in your database. For obvious reasons, you don't want to store unencrypted passwords in your database. You want to prevent evil hackers who have access to your database (because the sysadmin blundered) from stealing passwords used by your clients. Because hash functions are not at all reversible, you can store the password hashed with a function like `md5()` or `sha1()` so the evil hackers can't get the password in plain text.

6. Discuss Buffered Versus Unbuffered Queries

Buffered queries will retrieve the query results and store them in memory on the client side, and subsequent calls to get rows will simply spool through local memory. Buffered queries have the advantage that you can seek in them, which means that you can move the “current row” pointer around in the result set freely because it is all in the client. Their disadvantage is that extra memory is required to store the result set, which could be very large, and that the PHP function used to run the query does not return until all the results have been retrieved

Sections 2

Contain coding with its output..

7.FIND THE OUTPUT OF THE FOLLWING

```
Source History
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title></title>
5   </head>
6   <body>
7     <?php
8
9     $x = 7;
10
11     do { echo "x is: $x" ; $x++; }
12
13     while ($x<=5);
14
15     ?>
16     </body>
17
18   </html>
19
```

THE OUTPUT:

x is: 7

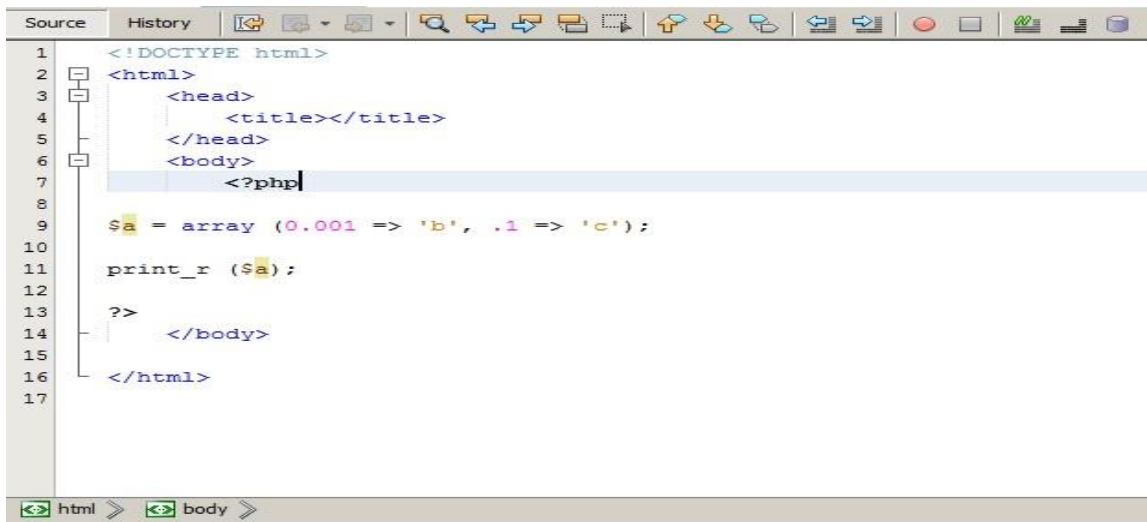


```
Source History
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title></title>
5   </head>
6   <body>
7     <?php
8
9     $a = 20;
10
11     function myFunction ($b) {
12       $a = 30 ;
13       global $a, $c;
14       return $c = ($b + $a);
15     }
16
17     print myFunction (40) + $c;
18
19     ?>
20   </body>

```

THE OUTPUT:

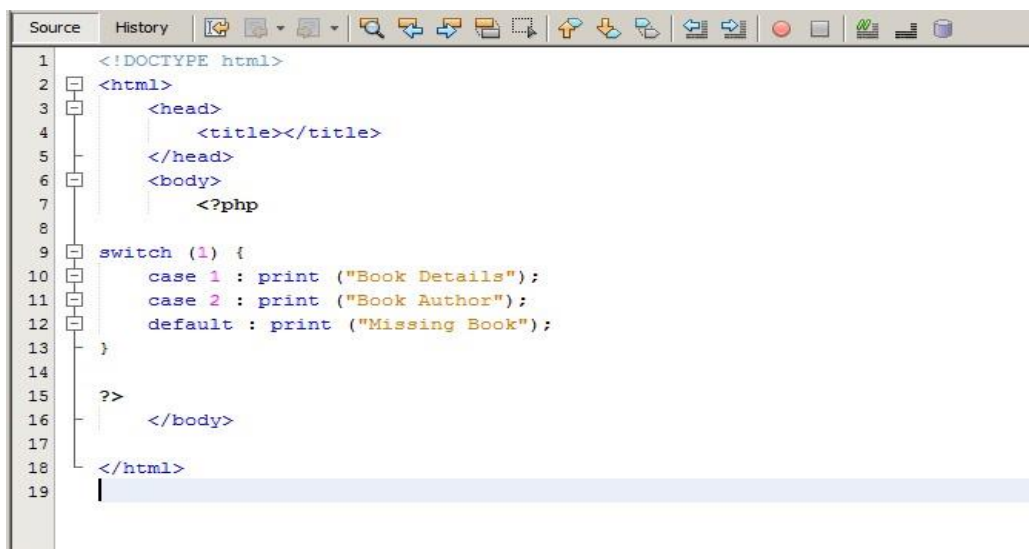
120



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title></title>
5   </head>
6   <body>
7     <?php
8
9     $a = array (0.001 => 'b', .1 => 'c');
10
11     print_r ($a);
12
13     ?>
14   </body>
15 </html>
16
17
```

THE OUTPUT:

Array ([0] => c)



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title></title>
5   </head>
6   <body>
7     <?php
8
9     switch (1) {
10      case 1 : print ("Book Details");
11      case 2 : print ("Book Author");
12      default : print ("Missing Book");
13    }
14
15    ?>
16  </body>
17 </html>
18
19
```

THE OUTPUT:

Book Details Book Author Missing Book

```
Source History [Icons]
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title></title>
5 </head>
6 <body>
7 <?php
8
9 $a = 'false';
10
11 if ($a)
12 {
13     { echo 'true';
14 }
15 else
16 { echo 'false';
17 }
18 ?>
19 </body>
20 </html>
21
```

THE OUTPUT:

true

Sections 3

Contain shorts coding in my point of view (try to find another example for CSS embedding with html).

8. Write a statement that loads the Apache Derby driver for the built-in embedded Java Database?

ANSWER:

Class.forName("org.apache.derby.jdbc.EmbeddedDriver");

9. Write a statement that creates a Connection object using a URL for the database called "employees" using the Apache Derby embedded database?

ANSWER:

```
Connection conn = DriverManager.getConnection("jdbc:derby:employees");
```

10. Write a php command that you must use to start using session variable.

```
Session_start ( );
```

11. Write a php variable that the student name is ahmed.

```
$_SESSION ['student'] = "ahmed" ;
```

12. What are the types of CSS Syntax Selector Strings

Single element type:

```
p { font-size:smaller; letter-spacing:1em }
```

Multiple element types:

```
h1,h2,h3,h4,h5,h6 { background-color:purple }
```

All element types:

```
* { font-weight:bold }
```

Specific elements by id:

```
#p1, #p3 { background-color:aqua }
```

Source anchor elements:

```
a:link { color:black }  
a:visited { color:yellow }  
a:hover { color:green }  
a:active { color:red }
```

Style sheets can be embedded directly in HTML document using style element.

```
<head>
  <title>InternalStyleSheet.html</title>
  <style type="text/css">
    h1, h2 { background-color:aqua }
  </style>
</head>
```

Element types that are descendants:

```
ul ol li { letter-spacing:1em }
```

Elements belonging to a style class:

```
#p4, .takeNote { font-style:italic }
```

Sections 4

Contain extra question..

13.For the Connection object, what is the difference between the createStatement, prepareStatement, and prepareCall methods?

ANSWER:

createStatement and prepareStatement allow the execution of SQL query strings in an ad-hoc manner. However, prepareStatement allows easy binding of user-provided parameters into that statement in a manner that is not subject to SQL injection security vulnerabilities. prepareCall allows the code to execute a stored procedure in the database that has been defined by the database administrator.

14. On what basis are elements in forms mapped into the \$_POST, \$_GET, or \$_REQUEST super global arrays?

ANSWER:

The HTML attribute “name” is used to map elements into the superglobals. For example, `<input type='text' id='foo' name='bar' />` would appear in \$_REQUEST using ‘bar’ as the index.

15. Explain the difference between the ‘==’ and the ‘===’ operators. Given an example of two variables (\$a and \$b) such that \$a == \$b is true, but \$a === \$b is false.

ANSWER:

The == operator employs type juggling to do the comparison. However, the === operator not only compares values, but also types (i.e. no juggling is permitted). For example, when \$a=0 and \$b='0', then \$a==\$b will be true, but \$a=== \$b is false because the types don’t match.

16. In a web application, how is the execution of JavaScript fundamentally different than PHP?

ANSWER:

JavaScript is executed in the browser on the user’s machine. On the other hand, PHP executes on the server and produces the HTML and JavaScript that is sent to the user’s machine.

17. How are the GET and POST methods of a web form different in how data is transmitted to the server?

ANSWER:

In a GET method, parameters from a web form are encoded as part of the URL (for example `http://mycompany.com/app/somePage.php?name=George&city=Columbus` indicates that two parameters – name and city – are given). In a POST method, the parameters are put inside the body of the HTTP request and are not visible in URL bar of the browser.

18. How are single-quote strings different from double-quote strings in PHP?

ANSWER:

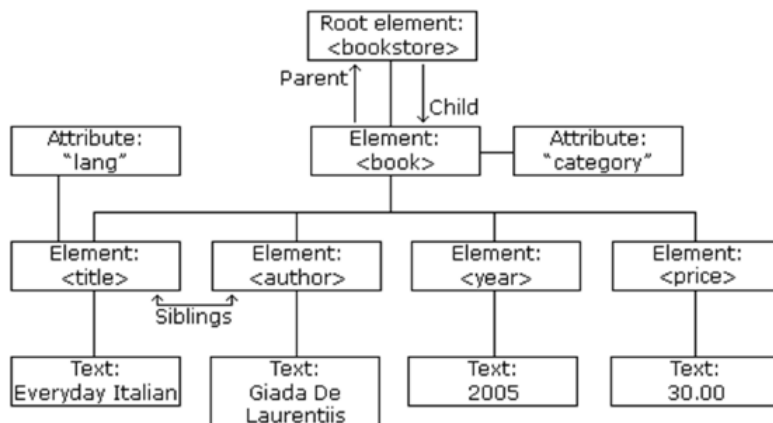
Single-quote strings do not perform variable interpolation or handle escape sequences. Double-quote strings are parsed and handle variable interpolation and escape sequences.

Sections 5

Contain long coding(check out the last revision and I hop that the long answer may contain the (XML assignment) and php revision array.

19. Create XML document to display the data of Book Store containing the books according to the following tree. Display the data of any 3 different books.

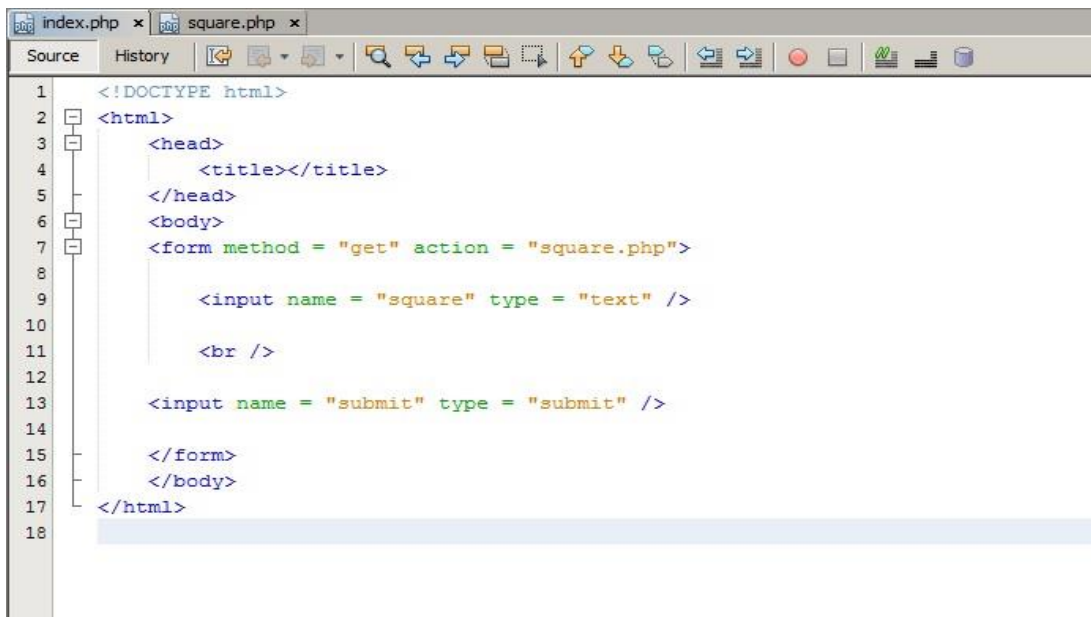
Tree structure for the Book Store as follow:



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
<book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>
<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
```

```
<price>29.99</price>
</book>
<book category="WEB">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>
<book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>
```

20. Make a web page that contain 2 pages which ask the user to enter any number and get the squared of this number using PHP?



The screenshot shows a web browser window with two tabs: 'index.php' and 'square.php'. The 'square.php' tab is active, displaying a form. The form has a text input field with the name 'square' and a submit button with the name 'submit'. The form is enclosed in a form tag with method 'get' and action 'square.php'. The browser's source code view is open, showing the HTML structure of the page. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title></title>
5   </head>
6   <body>
7     <form method = "get" action = "square.php">
8
9       <input name = "square" type = "text" />
10
11      <br />
12
13     <input name = "submit" type = "submit" />
14
15    </form>
16  </body>
17 </html>
18
```

```
index.php x square.php x
Source History
<?php
2
3 if (isset ($_GET ['submit']))
4
5 {
6     $x = $_GET ['square'];
7
8     echo $x * $x ;
9 }
10
11 else
12 {
13
14 }
15
16
```