Tutorial Letter 102/2/2018

Advanced Internet Programming

ICT3612

Semester 2

School of Computing

This tutorial letter contains important information about ICT3612 as well as Assignment 1 and 2

Define tomorrow.

UNISA university of south africa

Contents

1 INTRODUCTION 3
2 PURPOSE OF AND OUTCOMES FOR THE MODULE 3
2.1 Purpose 3
2.2 Outcomes 3
3 MODULE FORMAT: ICT3612 4
3.1 Fully online module 4
4 LECTURER, TUTOR AND CONTACT DETAILS 4
4.1 Lecturer 4
4.2 School of Computing contact details 4
4.3 University 4
4.4 Online discussion forum 4
4.5 E-mail 5
5 ASSESSMENT 5
5.1 Assessment plan 5
5.2 Unique assignment numbers and due dates of assignments 5
5.3 Submission of assignments 5
6 STUDY MATERIAL AND COURSE APPROACH 6
6.1 Study material/resources 6
6.2 Course approach 6
7 8
ASSIGNMENTS 9
7.1 Assignment 1 (chapters 13-18) Multiple Choice 9
7.2 Assignment 2 (chapters 19-23) Multiple Choice 25

1 INTRODUCTION

Welcome to ICT3612, Advanced Internet Programming. Please read this tutorial letter carefully.

In ICT2613 you studied chapters 1 -12 of the prescribed handbook. This semester, in ICT3612, you will study chapters 13-23. As with ICT2613, the *instructional objectives* of this course are *knowledge* and *application*. The formative assessment (examination) will test both knowledge and application.

Best wishes with your studies!
The Internet Programming Tuition Team

2 PURPOSE OF AND OUTCOMES FOR THE MODULE

2.1 Purpose

This module will be useful to provide an Internet programmer with the advanced knowledge, skills and values to analyse, design, develop and maintain Internet solutions in information systems of environments such as industrial processes systems, commerce and organizational information systems.

- The student will be able to analyse, design, develop and maintain advanced Internet applications.
- The student will be able to design, develop and maintain advanced web sites.
- Qualified students will able to design, develop and maintain advanced web-sites and provide long term principled decisions in a changing society to specific international standards of precision, robustness, reliability, cost-effectiveness, solution-specific within a specified time-frame and client-satisfaction.
- Students will be able to analyse, design and develop appropriate advanced web-based solutions in a changing environment.

2.2 Outcomes

Specific outcome 1: Demonstrate an understanding of and create dynamic web-based applications.
Assessment criteria: Web-based applications are dynamic according to industry standards.

Specific outcome 2: Build rich Internet applications.
Assessment criteria: The created Internet application adheres to completeness standards according to industry; the application enhances the presentation of

online content; web applications have the look and feel of desktop applications; applications are database-driven.

Specific outcome 3: Build real-life, industrial strength web-based applications.
Assessment criteria: Web-based applications are applicable in real-life and portray industrial strength; created applications will interact with pre-implemented server-side applications and web-servers.

3 MODULE FORMAT: ICT3612

3.1 Fully online module

All study material for this module will be available on *my*Unisa. It is thus very important that you register on *my*Unisa and access the module site on a regular basis.
Please consult with the *my*Studies @Unisa publication for more information on the activation of your *my*Life email address as well as obtaining access to the *my*Unisa module site.

4 LECTURER, TUTOR AND CONTACT DETAILS

4.1 Lecturer

To send an email to the course leader, please use the Course Contact option on *myUnisa*.

To contact the University, you should follow the instructions in the *myStudies@Unisa* brochure. Remember to have your student number available when you contact the University.

4.2 School of Computing contact details

Tel: +27 11 670 9200
E-mail: computing@unisa.ac.za

4.3 University

Please consult the *myStudies@Unisa* brochure for general Unisa contact details. Remember to use your student number when contacting the University.

4.4 Online discussion forum

Since the focus of this course is on Internet programming, the primary means of communication and support is done over the Internet. Specifically, we make use of an online discussion forum. Please make use of the official ICT3612 myUnisa forum. The purpose of the discussion forum is to provide students with a collaborative environment where they can openly ask questions en route to successful completion of the practical work. The advantage of using a forum is that students often experience the same problems, and when a query is

posted on the discussion forum, all students benefit from the discussion that follows.

4.5 E-mail

Please make use of the Course Contact option on myUnisa. This will prevent the chance of your email being marked as spam.

5 ASSESSMENT

5.1 Assessment plan

There are three summative assessments (two MCQ assignments and a practical assignment), and one formative assessment (examination) for this module.

The first two summative assessments (assignments) measure mastery of the *knowledge.*

Each assignment will contribute 10% towards your year mark. An important motivation for using MCQs is that it forces you to read through the handbook en route to gaining knowledge.
The third summative assessment consists of several PHP tasks that you have to code throughout the semester. You will then upload these tasks to a free web hosting server and submit the URL to us for marking (80 % of the year mark). The year mark will contribute 30% to the final mark.
The formative assessment (examination) will assess your mastery of the practical skills learned in completing the portfolio. It will be a closed book examination, and will contribute 70% towards your final mark.
It follows that you cannot afford to not complete the portfolio (Assignment 03) on your own!

5.2 Unique assignment numbers and due dates of assignments

Assignment number	Unique assignment number	Assignment due dates
1	824151	13 August
2	762562	03 September
3	865523	05 October

5.3 Submission of assignments

All assignments are submitted online through myUnisa

6 STUDY MATERIAL AND COURSE APPROACH

6.1 Study material/resources

- Tutorials letters 101 - 103.
- The handbook which was prescribed for ICT2613. ICT2613 covered chapters 1-12. ICT3612 covers chapters 13-23, inclusive of the knowledge gained from studying chapters 1-12.

6.2 Course approach

The course approach is similar to ICT2613. The prescribed handbook teaches all the skills you need to develop almost any type of database- driven web site with PHP and MySQL. The chapters gradually move from the simple to the complex as they show you how to work with:

- the PHP language
- the MySQL database
- the SQL statements for MySQL
- the PDO and mysqli extensions
- the MVC pattern
- debugging and exception handling
- sessions and cookies
- object-oriented PHP
- regular expressions and data validation
- SSL and secure connections
- authentication and login forms
- email and content from other sites (like YouTube)
- file uploads
- and more!

The content is presented in "paired pages" with the essential syntax, guidelines, and examples on the right page and the perspective and extra explanation on the left page. The in-chapter exercises on the right-hand side are used to explain and illuminate the theory which is presented on the left-hand pages. In this manner you add code to an in-chapter exercise application as you progress through the chapter. There is thus always a clear link between knowledge and the application of such knowledge – which is what sets this handbook apart from any other handbook. This approach helps you *learn faster by reading less*, and it makes an ideal reference format.

Most of the code and solutions to the exercises/applications are made available. The intention is for you to *try and complete the exercises/applications on your own,* and when you can't do so, to then make use of the solutions provided. In this manner you *learn* as you go along. The focus is thus on active and purposeful learning.

Step 1: Install server environment, editor and course files

- Appendix A (MAC) and Appendix B (Windows) cover installation the server environment, an editor and course files (book applications and source code). Note: on newer versions of Windows, do not install XAMPP to your C:/Program Files folder since you are prevented from writing to this folder.
- All the software is available on the course CD, so you do not have to download it as directed in these appendixes.
- Note that the server environment and editors are tools that may, depending on the configuration of your PC, present unique installation and usage problems that we cannot support. We suggest you Google or make use the forum – other people/students may have experienced the same problem/s.
- If you make use of a Unisa Computer Laboratory, note that another independent MySQLinstallation may be installed on the PC's. This may cause problems when you try to start MySQL through the XAMPP interface/panel. The solution is to check for another MySQL icon in the taskbar to the bottom right (in the past it was a small traffic robot icon, but we cannot be sure as laboratory installations change from year to year). Ensure this particular MySQL service is not running (stop it by right-clicking the icon -> WinNT -> Stop service) or through the Running Services application. Then restart the MySQL service from the XAMPP panel.
- Also note that if you run Skype, it will clash with XAMPP since the latter installation also makes use of port 80. Please Google for support on resolving this issue, or post your problem on the forum.
- The complete PHP manual (with a search option of PHP functions) is available as a windows help-file (in the resources folder: php_manual_en.chm) If you are using Windows XP SP2 or later, you need to "unblock" the file by right-clicking on it in Windows Explorer and selecting the "Properties" menu item, then clicking on the "Unblock" button (on Windows Vista this is within the "Security" options). Failure to unblock the documentation file may result in error messages including "Navigation to the webpage was canceled" due to Windows security restrictions.

Step 2: Work through each of the handbook chapters.

- Read the Introduction which explains how the book is structured. Note we provide all the software required on the CD, so there is no need to download it.
- In subsequent chapters, read the text presented on the left page, and where applicable, run/adapt the code presented on the right page (in-chapter exercises), which, if you have followed the steps in the book appendix, is available in the `books_app` folder in your root web folder.
- Complete the exercises presented at the end of each chapter. Starting code is available in the`ex_starts` folder.
- The solutions to all the exercises are available in the `ex_solutions` folder. This implies you can simply copy the solutions and run them, or just read through them - but as stated earlier, **you should use the solutions to learn**. Thus, try and complete an exercise on your own, and only use the solution when you get stuck. This route will enable you to learn and to answer some exam questions!

6.3 Study Plan

A semester is roughly 16 weeks. There are twelve chapters to work through, therefore we require you to work through a chapter (and the associated portfolio task/s for that chapter - see later) a week. Some chapters will go faster, some slower, but a chapter a week is feasible given the handbook and course approach. The portfolio tasks are spaced in line with this approach.

7

ASSIGNMENTS

7.1 Assignment 1 (chapters 13-18) Multiple Choice

Assignment number	1
Assignment due date	13 August
Unique assignment number	824151
Multiple Choice questions	50

Question 1

Which one of the following is the right way of defining a function in PHP?

1. `function functionName(parameters) { function body }`
2. `function { function body }`
3. `data type functionName(parameters) { function body }`
4. `functionName(parameters) { function body }`

Question 2

What output will the following PHP function produce?

```
<?php
     function calc($price, $tax = '')
     {
         $total = $price + ($price * $tax);
         echo "$total";
     }
     calc(42);
?>
```

1. 0
2. 42
3. 84
4. error

Question 3

Which of the following are valid function names?

i. `function()`
ii. `€()`
iii. `.function()`
iv. `$function()`

1. only ii)
2. none of the mentioned
3. all of the mentioned
4. i)

Question 4

What output will the following function produce?

```
<?php
    function b()
    {
        echo "b is executed";
    }
    function a()
    {
        b();
        echo "a is executed";
        b();
    }
    a();
?>
```

1. `b is executedb is executedb is executed`
2. `b is executeda is executedb is executed`
3. `b is executeda is executed`
4. `a is executed`

Question 5

What output will the following function produce?

```
<?php
    $op2 = "blabla";
    function foo($op1)
    {
        echo $op1;
        echo $op2;
    }
    foo("hello");
?>
```

1. `Hello`
2. `Helloblabla`
3. `Error`
4. `helloblablablabla`

Question 6

Which one of the following PHP functions can be used to build a function that accepts any number of arguments?

1. `get_argv()`
2. `get_argc()`
3. `func_get_argv()`
4. `func_get_argc()`

Question 7

What output will the following PHP code produce?

```
<?php
   function addFive($num)
   {
        $num += 5;
   }
   function addSix(&$num)
   {
        $num += 6;
   }
   $orignum = 10;
   addFive( $orignum );
   echo "Original Value is $orignum<br />";
   addSix( $orignum );
   echo "Original Value is $orignum<br />";
?>
```

1. `Original Value is 15`
   `Original Value is 21`
2. `Original Value is 21`
   `Original Value is 15`
3. `Original Value is 10`
   `Original Value is 16`
4. `Original Value is 10`
   `Original Value is 15`

Question 8

What output will the following PHP code produce?

```
<?php
     $fruits = array ("apple", "orange", "banana");
     echo (array_shift ($fruits));
     echo (array_shift ($fruits));
?>
```

1. `orangebanana`
2. `orangeorange`
3. `appleapple`
4. `appleorange`

Question 9

The practice of separating the user from the true inner workings of an application through well-known interfaces is known as…

1. polymorphism
2. encapsulation
3. abstraction
4. inheritance

Question 10

Which one of the following property scopes is not supported by PHP?

1. final
2. public
3. friendly
4. static

Question 11

Which one of the following can be used to instantiate an object in PHP assuming class name to be `Foo`?

1. `$obj = new $foo;`
2. `$obj = new foo ();`
3. `obj = new foo ();`
4. `$obj = new foo;`

Question 12

Which one of the following is the correct way to define a constant?

1. `constant PI = "3.1415";`
2. `const $PI = "3.1415";`
3. `const PI = '3.1415';`
4. `constant PI = '3.1415';`

Question 13

Which one of the following is the correct way to call a class constant, given that the class is `mathFunction`?

1. `echo PI;`
2. `echo mathFunction->PI;`
3. `echo mathFunction::PI;`
4. `echo mathFunction=PI;`

Question 14

Which one of the following is the correct way to invoke a method?

1. `object->methodName();`
2. `$object->methodName();`
3. `object->methodName();`
4. `object::methodName();`

Question 15

Which of the following is/are the correct way to declare a method?

i) function functionName() { function body }
ii) scope function functionName() { function body }
iii) method methodName() { method body }
iv) scope method methodName() { method body }

1. only i)
2. only iv)
3. i) and ii)
4. iii) and iv)

Question 16

Which method scope prevents a method from being overridden by a subclass?

1. Final
2. Abstract
3. protected
4. static

Question 17

Which one of the following functions is used to determine whether a class exists?

1. `exist()`
2. `exist_class()`
3. `__exist()`
4. `class_exist()`

Question 18

Which one of the following keywords is used to inherit a subclass into a superclass?

1. `extends`
2. `implements`
3. `inherit`
4. `include`

Question 19

In the PHP code given below, what is/are the properties?

```
<?php
    class Example
    {
        public $name;
        function Sample()
        {
            echo "This is an example";
        }
    }
?>
```

1. `echo “This is an example”;`
2. `class Example`
3. `public $name;`
4. `function sample()`

Question 20

Which keyword is used to refer to properties or methods within the class itself?

1. `private`
2. `public`
3. `protected`
4. `$this`

Question 21

Which keyword allows class members (methods and properties) to be used without needing to instantiate a new instance of the class?

1. `static`
2. `protected`
3. `final`
4. `private`

Question 22

Which one of the following statements is used to create a table?

1. `CREATE table_name (column_type column_name);`
2. `CREATE table_name (column_name column_type);`
3. `CREATE TABLE table_name (column_type column_name);`
4. `CREATE TABLE table_name (column_name column_type);`

Question 23

Which one of the following statements instantiates the mysqli class?

1. `mysqli = new mysqli()`
2. `$mysqli->new.mysqli()`
3. `$mysqli = new mysqli()`
4. `mysqli->new.mysqli()`

Question 24

What output will the following PHP code produce?

```
<?php
    define("GREETING","Hello you! How are you today?");
    echo constant("GREETING");
?>
```

1. GREETING
2. GREETING, Hello you! How are you today?
3. Hello you! How are you today?
4. "GREETING","Hello you! How are you today?"

Question 25

What output will the following PHP code produce?

```
<?php
    function b()
    {
        echo "b is executed";
    }
    function a()
    {
        b();
        echo "a is executed";
        b();
    }
    a();
?>
```

1. b is executedb is executedb is executed
2. b is executeda is executed
3. b is executeda is executedb is executed
4. a is executed

Question 26

What output will the following PHP code produce?

```
<?php
    function A1($x)
    {
        switch($x)
        {
        case 1:
            echo 'Case 1 was executed.';
            break;
        case 2:
            echo 'Case 2 was executed.';
            break;
```

```
            case 3:
                echo 'Case 3 was executed.';
                break;
            case 4:
                echo 'Case 4 was executed.';
                break;
            default:
                echo 'Default was executed.';
                break;

            }
        }
        A1(9);
    ?>
```

1. case 1 was executed
2. case 2 was executed
3. case 4 was executed
4. default was executed

Question 27

What output will the following PHP code produce?

```
<?php
    function colour()
    {
         $colors = array("red", "green", "blue", "yellow");
         foreach ($colors as $value)
         {
              echo "$value <br>";
         }
    }
    colour();
?>
```

1. red green blue yellow
2. red
   green
   blue
   yellow
3. red
   blue
   yellow
   green
4. redgreenblueyellow

Question 28

What output will the following PHP code produce?

```
<?php
    $line = "You like dogs. I hate dogs. We should marry."
    $sen = preg_split('/\./', $line);
    print_r($sen);
?>
```

1. Error You like dogs.
2. Array([0]=>You like dogs. I hate dogs. We should marry.)
3. Array([0]=>You like dogs [1]=>I hate dogs [2]=>We should marry[3]=>)
4. You like dogs. I hate dogs. We should marry.

Question 29

What is the Regular Expression matching zero or more specific characters?

1. x
2. *
3. #
4. &

Question 30

Which one of the following regular expression matches any string containing zero or one p?

1. p+
2. p*
3. p?
4. p#

Question 31

What output will the following PHP code produce?

```
<?php
        $username = "jasoN";
        if (ereg("([^a-z])",$username))
        echo "Username must be all lowercase!";
    else
        echo "Username is all lowercase!";
?>
```

1. Error
2. `Username must be all lowercase!`
3. `Username is all lowercase!`
4. No Output is returned

Question 32

Using SQL, select all the records from a table named "Persons" where the value of the column "FirstName" starts with an "a"?

1. `SELECT * FROM Persons WHERE FirstName='%a%'`
2. `SELECT * FROM Persons WHERE FirstName LIKE 'a%'`
3. `SELECT * FROM Persons WHERE FirstName='a'`
4. `SELECT * FROM Persons WHERE FirstName LIKE '%a'`

Question 33

Using SQL, select all the records from a table named "Persons" where the "FirstName" is "Peter" and the "LastName" is "Jackson"?

1. `SELECT FirstName='Peter', LastName='Jackson' FROM Persons`
2. `SELECT * FROM Persons WHERE FirstName='Peter' AND LastName='Jackson'`
3. `SELECT * FROM Persons WHERE FirstName<>'Peter' AND LastName<>'Jackson'`
4. `SELECT * FROM Persons WHERE FirstName='Peter' AND LastName<>'Jackson'`

Question 34

Using SQL, select all the records from a table named "Persons" where the "LastName" is alphabetically between (and including) "Hansen" and "Pettersen"?

1. `SELECT * FROM Persons WHERE LastName>='Hansen' AND LastName=<'Pettersen'`
2. `SELECT * FROM Persons WHERE LastName BETWEEN 'Hansen' AND 'Pettersen'`
3. `SELECT LastName>'Hansen' AND LastName<'Pettersen' FROM Persons`
4. `SELECT * FROM Persons WHERE LastName>'Hansen' AND LastName<'Pettersen'`

Question 35

Using SQL, which line will return all the records from a table named "Persons" sorted in descending order by "FirstName"?

1. `SELECT * FROM Persons SORT 'FirstName' DESC`
2. `SELECT * FROM Persons ORDER FirstName DESC`
3. `SELECT * FROM Persons ORDER BY FirstName DESC`
4. `SELECT * FROM Persons SORT BY 'FirstName' DESC`

Question 36

Using SQL, which line will insert a new record into the "Persons" table?

1. `INSERT VALUES ('Jimmy', 'Jackson') INTO Persons`
2. `INSERT INTO Persons VALUES ('Jimmy', 'Jackson')`
3. `INSERT ('Jimmy', 'Jackson') INTO Persons`
4. `INSERT VALUES INTO Persons VALUES ('Jimmy', 'Jackson')`

Question 37

Using SQL, which line will insert "Olsen" as the "LastName" in the "Persons" table?

1. `INSERT ('Olsen') INTO Persons (LastName)`
2. `INSERT INTO Persons ('Olsen') INTO LastName`
3. `INSERT INTO Persons (LastName) VALUES ('Olsen')`
4. `INSERT VALUE 'Olsen' INTO LastName`

Question 38

Using SQL, which line will change "Hansen" into "Nilsen" in the "LastName" column in the Persons table?

1. `MODIFY Persons SET LastName='Hansen' INTO LastName='Nilsen`
2. `UPDATE Persons SET LastName='Hansen' INTO LastName='Nilsen'`
3. `MODIFY Persons SET LastName='Nilsen' WHERE LastName='Hansen'`
4. `UPDATE Persons SET LastName='Nilsen' WHERE LastName='Hansen'`

Question 39

Using SQL, how can you delete the records where the "FirstName" is "Peter" in the Persons Table?

1. `DELETE ROW FirstName='Peter' FROM Persons`
2. `DELETE FirstName='Peter' FROM Persons`
3. `DELETE FROM Persons WHERE FirstName = 'Peter'`
4. `DELETE 'Peter' FROM Persons`

Question 40

Using SQL, return the number of records in the "Persons" table?

1. `SELECT COUNT() FROM Persons`
2. `SELECT COLUMNS(*) FROM Persons`
3. `SELECT COUNT(*) FROM Persons`
4. `SELECT COLUMNS() FROM Persons`

The next nine (9) questions refer to the class below:

```
class Vegetable {

      public $edible;
      public $color;
      private $name;

   public function __construct($edible,$color,$name)
      {
        …
      }
      function isEdible()
      {
            …
      }
      function whatColor()
      {
            …
      }
      function display()
      {
            …
      }

} // end of class Vegetable
```

Question 41

Which of the following creates a blue edible `Vegetable` object?

1. `$veggie = new Vegetable(true, "blue");`
2. `$veggie = Vegetable(true, "blue");`
3. `$veggie = new Vegetable(edible, "blue");`
4. `veggie = new Vegetable(true, "blue");`

Question 42

A program creates a blue edible `Vegetable` object. Which of the following calls the display method?

1. `veggie->display();`
2. `$veggie.display();`
3. `$veggie->display;`
4. `$veggie->display();`

Question 43

Which of the following will display the color of a `Vegetable` object?

1. `echo veggie->whatColor();`
2. `echo $veggie.whatColor();`
3. `echo $veggie->whatColor();`
4. `display $veggie->whatColor();`

Question 44

Assume the following declaration

```
$veggie = new Vegetable(true, "blue", "eggplant");
```

What output will be produced by the statements below?

```
foreach ($veggie as $veg => $value) {
    echo "<li>$veg = $value</li>";
}
```

1. `edible = 1`
   `color =`
   `name =`

2. `edible = 1`
   `color = blue`
   `name = eggplant`

3. `edible = 1 color = blue  name = eggplant`
4. `edible = 1 color = name =`

Question 45

What code will implement the method `isEdible()`?

1. `function isEdible()`
`{`
`  return this->edible;`
`}`
	2. `function isEdible()`
	`{`
	`  return $this=edible;`
	`}`

3. `function isEdible()`
`{`
`  $this->edible;`
`}`
	4. `function isEdible()`
	`{`
	`  return $this->edible;`
	`}`

Question 46

Which of the following is a valid destructor of the class name `Vegetable`?

1. `public __destruct()`
2. `public function __destruct`
3. `public function __destruct()`
4. `public function destruct()`

Question 47

The subclass `Spinach` extends the base class `Vegetable`. The `Spinach` class constructor can call the `Vegetable` class with the statement _____.

1. `parent __construct();`
2. `parent::__construct();`
3. `super::__construct();`
4. `construct(true, "green", "eggplant");`

Question 48

Which of the following statements will determine if an object named `broccoli` is an instance of the class `Vegetable`?

1. `get_class($broccoli, 'Vegetable')`
2. `class_exists('Vegetable')`
3. `property_exists($broccoli, 'Vegetable')`
4. `is_a($broccoli, 'Vegetable')`

Question 49

Which of the following statements will determine if an object named `broccoli` has the property `color`?

1. `property_exists(broccoli, color)`
2. `property_exists(broccoli, 'color')`
3. `property_exists($brocolli, 'color')`
4. `property_exists($broccoli, color)`

Question 50

Indicate the line number in the code below where the compiler should find an error?

```
1   <?php
2       class Pet {
3           private $name;
4
5           public function __construct ($name){
6               $this->name = $name;
7           }
8           function getName(){
9               return $this->name;
10          }
11          function setName($value){
12              return $this->name = $value;
13          }

14          function display(){
15              echo 'The name of my pet is '.$this->name;
16          }
17      }
18      $pet1 = Pet('Ben');
19      $pet1->display();
20  ?>
```

1. line 3
2. line 5
3. line 12
4. line 18

7.2 Assignment 2 (chapters 19-23) Multiple Choice

Assignment number	2
Assignment due date	03 September
Unique assignment number	762562
Multiple Choice questions	45

Question 1

When you use a prepared SQL statement with PDO, you need to _____ the parameters to variable values before you can run the statement.

1. attach
2. connect
3. join
4. bind

Question 2

After you run a prepared SQL statement with PDO, which method of the `PDOStatement` class will get an array that corresponds with the rows in the result set?

1. `fetchall()`
2. `fetch()`
3. `rowCount()`
4. `bindValue()`

Question 3

When you use the procedural interface of `mysqli`, you use _____ to do the database operations.

1. operators
2. variables
3. functions
4. objects

Question 4

One disadvantage of using `mysqli` is that

1. it doesn't provide an object-oriented interface
2. it doesn't provide for prepared statements
3. it can only be used with MySQL
4. it only provides a procedural interface

Question 5

The prepare method of a PDO object

1. assigns the values to the parameters in a prepared statement
2. must be run before values are assigned to the parameters in a prepared statement
3. retrieves the rows of a prepared statement
4. must be run after values are assigned to the parameters in a prepared statement

Question 6

The `lastInsertID` method of the PDO class

1. sets the ID for the last record that has been inserted into a database
2. gets the ID for the last record with an auto-increment ID that has been inserted into a database
3. sets the ID for the last record of several that are going to be inserted into a database
4. gets the ID for the last record that has been inserted into a database

Question 7

When you use the exception error mode with PDO objects, you can

1. use the `errorCode` and `errorInfo` methods to find out whether an error occurred
2. use the `errorCode` and `errorInfo` properties to find out whether an error occurred
3. use try-catch statements to handle the exceptions that occur
4. use try-catch statements to throw exceptions

Question 8

In the object-oriented interface for `mysqli`, you use the query method to run a SELECT statement. In the procedural interface, you use this function:

1. `mysqli_query`
2. `query_mysqli`
3. `execute_mysqli`
4. `mysqli_execute`

Question 9

When you develop a content management system, you can use PHP's string-handling functions to parse the user entries and apply the _____ for formatting the content in the browser.

1. SQL
2. PDO
3. HTML
4. Utility file

Question 10

In web site that uses the MVC pattern, it's common to store the files for each application within the site in a single _____.

1. HTML file
2. utility file
3. include file
4. directory

Question 11

When you use PHP to format the data that's entered into a content management system, you

1. remove all HTML tags from the data
2. never convert special characters to HTML character entities
3. add HTML tags to the data wherever necessary
4. always convert special characters to HTML character entities

Question 12

Include files may include all but one of the types of files that follow. Which one is it?

1. controller
2. utility
3. model
4. view

Question 13

To set the include path for a web site, you can use the

1. `include_path` function
2. `include_path` method
3. `set_include_path` function
4. `set_include_path` method

Question 14

For a web site that implements the MVC pattern, there is usually one controller file

1. for each application
2. for each model
3. for each view
4. for each user

Question 15

Secure Sockets Layer (SSL) is a _____ that's used by the Internet that allows clients and servers communicate over a secure connection.

1. Connection
2. controller
3. certificate
4. authentication

Question 16

The URL that connects to `www.company.com` over a secure connection is _____.

1. www.company.com/secure
2. https://www.company.com/secure
3. http://www.company.com
4. https://www.company.com

Question 17

When SSL is used, the browser _____ all data that is sent to the server.

1. validates
2. moves
3. digest
4. encrypts

Question 18

The process of determining whether a server or client is who and what it claims to be is called _____.

1. accreditation
2. authentication
3. certification
4. authorization

Question 19

When you use passwords to determine whether a user is authorized to use an application, the passwords are normally _____ before they are stored in the database.

1. validated
2. confirmed
3. encrypted
4. endorsed

Question 20

PHP provides _____ that let you encrypt data before it's stored in a database and decrypt data after it's extracted from the database.

1. validation
2. confirmation
3. functions
4. authorization

Question 21

Before a server uses SSL to transmit data to a client, it

1. authorizes the client
2. provides a digital secure certificate to the client
3. decrypts all data
4. confirms the identity of the client

Question 22

A digital secure certificate is provided

1. for a fee from a registration authority
2. free from a certificate authority
3. for a fee from a certificate authority
4. free from a registration authority

Question 23

The maximum SSL strength for an application's connections is determined by

1. the SSL methods that you use
2. the certification authority
3. the registration authority
4. the digital secure certificate

Question 24

Once you establish a secure connection, the application

1. returns to a non-secure connection for subsequent requests
2. uses absolute URLs to continue using the secure connection
3. continues to use a secure connection until an absolute URL specifies a non-secure connection
4. continues to use a secure connection until a relative URL specifies a non-secure connection

Question 25

The `$_SERVER` array contains data that can be used to

1. redirect a page to a secure connection
2. determine whether the data has been encrypted
3. determine the SSL strength
4. forward a page to a secure connection

Question 26

When you use form-based authentication,

1. the username and password have to be sent for every protected page.
2. the username and password are automatically encrypted and decrypted.
3. the programmer writes the code for the authentication process.
4. the browser automatically displays a dialog box for the username and password.

Question 27

When you use form-based authentication, you need to include two files at the top of each protected page. The first gets a secure connection, and the second

1. asks for the username and password.
2. compares the password in the `$_SESSION` array with the password in the database.
3. decrypts the username and password that's in the database.
4. tests whether the `$_SESSION` array indicates that the user has been validated.

Question 28

You usually don't need to decrypt passwords, because

1. passwords aren't normally encrypted
2. you can compare a user's password with a database password in encrypted form
3. the SHA-1 function doesn't provide for decryption
4. form-based authentication doesn't require encryption

Question 29

One way to read data from a file is to use a PHP function that returns the entire file as a _____.

1. blog
2. directory
3. document
4. string

Question 30

A CSV file is a file in which data items are separated by _____.

1. Spaces
2. commas
3. semi-colons
4. new lines

Question 31

When you use a function to read a CSV file, the data items are returned in a/an _____.

1. string
2. directory
3. array
4. integer

Question 32

When you use PHP's `getimagesize` function, it returns the _____, width, and height of the specified image.

1. type, width, and name of the specified image
2. type, name, and height of the specified image
3. type, width, and height of the specified image
4. type, width, name , and height of the specified image

Question 33

The code that follows saves the data that was in the `$names` array as

```
$names = array('joelmurach', 'rayharris', 'mikemurach');
$names = implode("\n", $names);
file_put_contents('usernames.txt', $names);
```

1. a string with no delimiters
2. a string with data items separated by commas
3. a string with data items separated by new line characters
4. a string with data items separated by tabs

Question 34

To create a browser control for uploading a file, you code

1. an HTML input tag with the type attribute equal to "file"
2. an HTML input tag with the type attribute equal to "upload"
3. an HTML text box followed by a button that says "Browse..."
4. an HTML text box followed by a button that says "Upload"

Question 35

To save a file that you've uploaded to a permanent location, you can use this PHP function:

1. `copy`
2. `copy_uploaded_file`
3. `fwrite`
4. `move_uploaded_file`

Question 36

When you use PHP's function for resizing an image, you need to pass arguments that specify all but one of the following. Which one is it?

1. the x and y coordinates in the old image that tell where the copying should start
2. the x and y coordinates in the new image that tell where the copying should start
3. the path for the new image
4. the height and width of the new image

Question 37

The `filesize()` function returns the file size in ___.

1. Gigabytes
2. Kilobytes
3. Bytes
4. Bits

Question 38

Which one of the following function is capable of reading a file into an array?

1. `file()`
2. `arrfile()`
3. `arr_file()`
4. `file_arr()`

Question 39

Which one of the following functions is capable of reading a line from a file?

1. `fgets()`
2. `fget()`
3. `fileget()`
4. `filegets()`

Question 40

Which one of the following functions outputs string data to the specified file?

1. `filewrite()`
2. `fwrite()`
3. `filewrites()`
4. `fwrites()`

Question 41

Which one of the following functions is capable of reading a file into a string variable?

1. `file_contents()`
2. `file_get_contents()`
3. `file_content()`
4. `file_get_content()`

Question 42

Which one of the following functions is capable of reading a specific number of characters from a file?

1. `fread()`
2. `fget()`
3. `fileget()`
4. `filegets()`

Question 43

Which one of the following functions reads a line of comma-separated values and returns them in an array?

1. `fileget()`
2. `fgetcsv()`
3. `fileread()`
4. `fget()`

Question 44

How many items are available in the $_FILES array?

1. 2
2. 3
3. 4
4. 5

Question 45

Which one of the following functions reads a file and echoes it to the web page.

1. `filewrite()`
2. `rfile()`
3. `fileread()`
4. `readfile()`

©
UNISA
2018