



UNIVERSITY OF
WESTMINSTER

BIS Design and Development

5BUIS005C

Module Leader's Name: - Ms. Abarnah Kirupananda

Harithroo Wijayawardhana

W1742291

2018531

Part A

Task 1 create classes

Created the main class "Auction_item" and three sub classes "Furniture", "Book" and "Painting". Attributes of the main class are: "Id", "Seller", "Type", "Description", "AskingPrice", "ClosingDate", "CurrentBid", "Bidhistory". There are few attributes to sub classes since they are unique to those classes. Class "Furniture" has "Dimension". Class "Book" has "Author" and "Title". Class "Painting" has "Painter", "Year" and a private attribute "reservedPrice". Except for "reservedPrice" in "Book" class, all other attributes are public. I used the construct method to create new objects. There are 7 functions in the main class "TotalPricePaid()", "buyersfee()", "NetAmountSellersReceive()", "YourProfit()", "sold_in_askingprice()", "addtohistory", "printhishistory" and another function in "painting" class "compare". I calculated the buyer's premium separately "buyersfee" and added it to item price in another function "TotalPricePaid()" to get the total price.

```
1. class Auction_item
2.     { //main class
3.         var $Id;
4.         var $Seller;
5.         var $Type;
6.         var $Description;
7.         var $AskingPrice;
8.         var $ClosingDate;
9.         var $CurrentBid;
10.        var $Bidhistory;
11.
12.        function TotalPricePaid()
13.        {
14.            $fee = $this->buyersfee();
15.            $Total_Price = $this->AskingPrice + $fee;
16.            return $Total_Price;
17.        }
18.        function buyersfee()
19.        {
20.            $fee = $this->AskingPrice * 10 / 100; // (buyers premium)
21.            return $fee;
22.        }
23.        function NetAmountSellersReceive()
24.        {
25.            $Net_Amount_Sellers_Receive = $this-
26.            >AskingPrice - 10; // seller recieves the askingprice minus £10(fee for the website fro
27.            m seller/ sellers premium)
28.            return $Net_Amount_Sellers_Receive;
29.        }
30.        function YourProfit()
31.        {
32.            $fee = $this->AskingPrice * 10 / 100; // fee for the website from buyer
33.            $Your_Profit = $fee + 10; //profit to the website // 10 is the fee for the
34.            website from seller
35.            return $Your_Profit;
36.        }
37.        function sold_in_askingprice()
38.        {
39.            $Total_Price = $this->TotalPricePaid();
40.            $NetAmountSellersReceive = $this->NetAmountSellersReceive();
```

```

39.         $YourProfit = $this->YourProfit();
40.         $buyersfee = $this->buyersfee();
41.         return "Product ID " . $this->Id . ", Asking Price £" . $this->AskingPrice . ", Buyers premium £" . $buyersfee . ", sellers premium £10. <br>."Total price: £".$Total_Price." <br> Net amount the seller receives: £".$NetAmountSellersReceived." <br> Profit for the site: £".$YourProfit."<br>"; //message
42.     }
43.
44.     function addtohistory($date, $bid)
45.     { //function to add bidding records to the bid history
46.         $this->Bidhistory[$date] = $bid;
47.     }
48.
49.     function printhistory()
50.     { //function to print the bid history
51.         echo "item Id " . $this->Id . " the following bids are made:";
52.         echo "<table border='1'><tr><th>Date</th><th>Bid</th></tr>";
53.         foreach ($this->Bidhistory as $key => $value) {
54.             echo "<tr><td>".$key."</td><td> £".$value."</td></tr>";
55.         }
56.         echo "</table>";
57.     }
58. }
59.
60. class Furniture extends Auction_item
61. { //sub class
62.     var $Dimention; // specific details for the item
63.     // construct method
64.     function __construct($Dimention, $Id, $Seller, $Type, $Description, $AskingPrice, $ClosingDate, $CurrentBid, $Bidhistory)
65.     {
66.         $this->Dimention = $Dimention;
67.         $this->Id = $Id;
68.         $this->Seller = $Seller;
69.         $this->Type = $Type;
70.         $this->Description = $Description;
71.         $this->AskingPrice = $AskingPrice;
72.         $this->ClosingDate = $ClosingDate;
73.         $this->CurrentBid = $CurrentBid;
74.         $this->Bidhistory = $Bidhistory;
75.     }
76. }
77. class Book extends Auction_item
78. { //sub class
79.     var $Author; // specific details for the item
80.     var $Title; // specific details for the item
81.     // construct method
82.     function __construct($Author, $Title, $Id, $Seller, $Type, $Description, $AskingPrice, $ClosingDate, $CurrentBid, $Bidhistory)
83.     {
84.         $this->Author = $Author;
85.         $this->Title = $Title;
86.         $this->Id = $Id;
87.         $this->Seller = $Seller;
88.         $this->Type = $Type;
89.         $this->Description = $Description;
90.         $this->AskingPrice = $AskingPrice;
91.         $this->ClosingDate = $ClosingDate;
92.         $this->CurrentBid = $CurrentBid;
93.         $this->Bidhistory = $Bidhistory;
94.     }

```

```

95.     }
96.     class Painting extends Auction_item
97.     { //sub class
98.         var $Painter; // specific details for the item
99.         var $Year; // specific details for the item
100.        private $reservedPrice; // spesific details for the item (private)
101.        // construct method
102.        function __construct($Painter, $Year, $reservedPrice, $Id, $Seller, $Type
e, $Description, $AskingPrice, $ClosingDate, $CurrentBid, $Bidhistory)
103.        {
104.            $this->Painter = $Painter;
105.            $this->Year = $Year;
106.            $this->reservedPrice = $reservedPrice;
107.            $this->Id = $Id;
108.            $this->Seller = $Seller;
109.            $this->Type = $Type;
110.            $this->Description = $Description;
111.            $this->AskingPrice = $AskingPrice;
112.            $this->ClosingDate = $ClosingDate;
113.            $this->CurrentBid = $CurrentBid;
114.            $this->Bidhistory = $Bidhistory;
115.        }
116.
117.        public function compare()
118.        { // function for selling or not selling message
119.            if ($this->reservedPrice < $this->CurrentBid) {
120.                return "Painting '" . $this-
>Description . "' will be sold<br> "; // message
121.            } else {
122.                return "Painting '" . $this-
>Description . "' will remain unsold<br> "; // message
123.            }
124.        }
125.    }

```

Task 2 Create objects

Created objects using construct method.

```

1. // creating objects with construct method
2. $Furniture = new Furniture("85 cm X 35 cm X 52", "1", "Peter", "Furniture", "Coffee
table", "85", "31/03/2020", "60", array()); // information about the item and a empty
array is given to later add the bid history
3. $Book = new Book("William Shakespeare", "Macbeth", "2", "Ann", "Ann", "Play drama",
"15", "25/03/2020", "7", array()); // information about the item and a empty array is
given to later add the bid history
4. $Painting1 = new Painting("Hannah Fairfield", "1839", "1800", "3", "Helen", "Painti
ng", "Oil on Canvas", "3000", "30/04/2020", "850", array()); // information about the i
tem and a empty array is given to later add the bid history
5. $Painting2 = new Painting("Callypso", "2015", "85", "4", "Peter", "Furniture", "Pop
Art", "400", "30/04/2020", "420", array()); // information about the item and a empty
array is given to later add the bid history
6. $Painting3 = new Painting("Goya", "1800", "9500", "5", "Wendy", "Painting", "Spanis
h School", "15000", "31/05/2020", "10000", array()); // information about the item and
a empty array is given to later add the bid history

```

Task 3 call functions

When calling the functions, I called the function "sold_in_askingprice()" that prints a message and call the functions that need to be printed.

```
1. echo "(3) case 1<br>";
2.
3.     echo "<br>1.1<br> " . $Furniture->sold_in_askingprice();
4.     echo "<br>1.2<br> " . $Book->sold_in_askingprice();
5.     echo "<br>1.3<br> " . $Painting1->sold_in_askingprice();
6.     echo "<br>1.4<br> " . $Painting2->sold_in_askingprice();
7.     echo "<br>1.5<br> " . $Painting3->sold_in_askingprice();
```

Task 4 reserved price

Created the function "compare" to compare the two prices of "CurrentBid" and "reservedprice" to decide if the painting will be sold or will remain unsold.

```
1. class Painting extends Auction_item
2. { //sub class
3.     var $Painter; // specific details for the item
4.     var $Year; // specific details for the item
5.     private $reservedPrice; // specific details for the item (private)
6.     // construct method
7.     function __construct($Painter, $Year, $reservedPrice, $Id, $Seller, $Type, $Description, $AskingPrice, $ClosingDate, $CurrentBid, $Bidhistory)
8.     {
9.         $this->Painter = $Painter;
10.        $this->Year = $Year;
11.        $this->reservedPrice = $reservedPrice;
12.        $this->Id = $Id;
13.        $this->Seller = $Seller;
14.        $this->Type = $Type;
15.        $this->Description = $Description;
16.        $this->AskingPrice = $AskingPrice;
17.        $this->ClosingDate = $ClosingDate;
18.        $this->CurrentBid = $CurrentBid;
19.        $this->Bidhistory = $Bidhistory;
20.    }
21.
22.    public function compare()
23.    { // function for selling or not selling message
24.        if ($this->reservedPrice < $this->CurrentBid) {
25.            return "Painting " . $this->Description . " will be sold<br> "; // message
26.        } else {
27.            return "Painting " . $this->Description . " will remain unsold<br> "; // message
28.        }
29.    }
30. }
```

Task 5 Bid history

I used the function “addtohistory” to add bids to the bid history and function “printhistory” to print the history

Code for the two functions used to add bids to the history and print the history

```
1. function addtohistory($date, $bid)
2.     { //function to add bidding records to the bid history
3.         $this->Bidhistory[$date] = $bid;
4.     }
5.
6.     function printhistory()
7.     { //function to print the bid history
8.         echo "item Id ". $this->Id ." the following bids are made:";
9.         echo "<table border='1'><tr><th>Date</th><th>Bid</th></tr>";
10.        foreach ($this->Bidhistory as $key => $value) {
11.            echo " <tr><td>".$key."</td><td> £".$value."</td></tr>";
12.        }
13.        echo "</table>";
14.    }
```

Code to add bids to the bid history and printing the history

```
1. echo "<br>(5) case 3<br>";
2.
3.     echo $Furniture->addtohistory('25 January 2020', '20');
4.     echo $Furniture->addtohistory('31 January 2020', '28');
5.     echo $Furniture->addtohistory('01 February 2020', '35');
6.     echo $Furniture->addtohistory('10 February 2020', '45');
7.     echo $Furniture->addtohistory('12 February 2020', '60');
8.     echo $Painting1->addtohistory('20 January 2020', '550');
9.     echo $Painting1->addtohistory('27 January 2020', '600');
10.    echo $Painting1->addtohistory('15 February 2020', '850');
11.    echo "<br>3.1<br>";
12.    echo $Furniture->printhistory();
13.    echo "<br>3.2<br>";
14.    echo $Painting1->printhistory();
```

Part B

private

credentials.php

```
1. <?php
2. define("DB_SERVER", "localhost");
3. define("DB_USER", "root");
4. define("DB_PASS", "");
5. define("DB_NAME", "cw2partb");
6. ?>
```

database_functions.php

```
1. <?php
2. // seperate database from code
3. require("credentials.php");
4. function dbconnect() {
5.     $database=new mysqli(DB_SERVER, DB_USER, DB_PASS, DB_NAME);
6.     validate_connection($database);
7.     return $database;
8. }
9. function validate_connection($database) {
10.     if ($database->connect_errno==0){
11.     }
12.     if ($database->connect_errno>0){
13.         $msg="database connection failed: <br>";
14.         $msg.=$database->connect_error;
15.         // $msg = "database connection failed: ".$database->connect_error;
16.         echo $msg;
17.         echo "<br>";
18.     }
19. }
20. ?>
```

initialize.php

```
1. <?php
2. //create the database function
3. require("database_functions.php");
4. require("class_Tutors.php");
5. $db=dbconnect();
6. Tutor::set_database($db);
7. ?>
```

class_tutors.php

```
1. <?php
2. require_once('class_database.php');
3. class Tutor extends Database
4. {
5.     static public $table_name="tutors";
6.     static protected $db_columns = ['ID', 'Name', 'Email', 'Phone_number', 'Qualifications', 'Subjects_to_teach', 'Rate_per_hour'];
7.
8.     public $ID;
```

```

9.     public $Name;
10.    public $Email;
11.    public $Phone_number;
12.    public $Qualifications;
13.    public $Subjects_to_teach;
14.    public $Rate_per_hour;
15.
16. }
17. ?>

```

class_database.php

```

1. <?php
2. class Database {
3.
4.     static protected $database;
5.     static protected $table_name = "";
6.     static protected $columns;
7.     static public function set_database($database) {
8.         self::$database = $database;
9.     }
10.    static public function find_by_sql($sql){
11.        //echo $sql;
12.        $results = self::$database->query($sql);
13.        if(!$results) {                                     //add valuation of the quer
y succeded or failed
14.            exit("Database query failed. ");
15.        }
16.        $object_array=[];
17.        while($record = $results-
>fetch_assoc()) { //get the first row as an array and create an object
18.            $object_array[] = static::instantiate($record);
19.        }
20.        return $object_array;
21.    }
22.    static public function find_all() {
23.        $sql = "SELECT * FROM " . static::$table_name; //why not self::$table_name?
??
24.        return static::find_by_sql($sql);
25.    }
26.    static public function find_by_id($ID) {
27.        $sql = "SELECT * FROM " . static::$table_name . " WHERE ID='" . static::$da
atabase->escape_string($ID) . "'"; //escape the string
28.        $result = static::find_by_sql($sql);
29.        if(empty($result)){
30.            return array_shift($result);
31.        } else {
32.            echo "empty";
33.        }
34.    }
35.    static protected function instantiate($record) {
36.        $object = new static;
37.        foreach($record as $property => $value) {
38.            if(property_exists($object, $property)){
39.                $object->$property=$value;
40.            }
41.        }
42.        return $object;
43.    }
44.    static public function verify_user($loginname, $password){

```



```

45.         $sql = "SELECT * FROM " . static::$table_name . " WHERE loginname='$log
inname' AND password='$password' "; //escape the string
46.         $result = static::find_by_sql($sql);
47.         return array_shift($result);
48.     }
49.     public function update() {
50.         $attributes = $this->attributes();
51.         $attribute_pairs=[];
52.         foreach($attributes as $key=>$value) {
53.             $attribute_pairs[] = "{$key}='{$value}'";
54.         }
55.         $sql = "UPDATE " . static::$table_name . " SET ";
56.         $sql .= join(', ', $attribute_pairs);
57.         $sql .= " WHERE ID=' " . static::$database->escape_string($this-
>ID) . " '"; //
58.         $results = static::$database->query($sql); //
59.         return $results;
60.     }
61.     public function create() {
62.         $attributes = $this->attributes();
63.         $sql = "INSERT INTO " . static::$table_name . " (";
64.         $sql .= join(', ', array_keys($attributes));
65.         $sql .= ") VALUES (' ";
66.         $sql .= join("'", array_values($attributes));
67.         $sql .= " ' )";
68.         $results = static::$database->query($sql);
69.         if($results) {
70.             $this->ID = static::$database->insert_id;
71.         }
72.         return $results;
73.     }
74.     public function delete() {
75.         $sql = "DELETE FROM " . static::$table_name . " WHERE ID=' " . $this-
>ID . " ' LIMIT 1";
76.         $results = static::$database->query($sql);
77.         return $results;
78.     }
79. // $attributes are the properties that have the database columns excluding 'ID'
80.     public function attributes() {
81.         $attributes=[];
82.         foreach(static::$db_columns as $column) {
83.             if($column == 'ID') { continue; }
84.             $attributes[$column] = $this->$column;
85.         }
86.         return $attributes;
87.     }
88.     public function merge_attributes($args=[]) {
89.         foreach($args as $key=>$value) {
90.             if(property_exists($this, $key)) {
91.                 $this->$key = $value;
92.             }
93.         }
94.     }
95. }
96. ?>

```

Public

Index.php

```

1. <html>
2.
3. <head>
4.     <title>Online Academy</title>
5.     <link rel=stylesheet type=text/css href="st.css?v=<?php echo time();?>">
6. </head>
7.
8. <body>
9.     <?php
10.     echo "<h1>Online Academy</h1>";
11.     echo "<nav><h3>";
12.     echo "<a href='index.php'>Home</a> | <a href='Our tutors.php'>Our Tutors</a> | <a href='add.php'>Add tutor</a>";
13.     echo "</h3></nav>";
14.     echo "<h2>Welcome</h2>";
15.     echo "'Online academy' is a website made for a second year BIS design and development course work at IIT. Click on '<u>Our Tutors</U>' for more info";
16.     ?>
17. </body>
18.
19. </html>

```

Our tutors.php

```

1. <html>
2.
3. <head>
4.     <title>Our tutors</title>
5.     <link rel=stylesheet type=text/css href="st.css?v=<?php echo time();?>">
6. </head>
7.
8. <body>
9.     <?php
10.     echo "<h1>Online Academy</h1>";
11.     echo "<nav><h3>";
12.     echo "<a href='index.php'>Home</a> | <a href='Our tutors.php'>Our Tutors</a> | <a href='add.php'>Add tutor</a>";
13.     echo "</h3></nav>";
14.     include '../private/initialize.php'; // initialize the web site
15.     $varray = Tutor::find_all();
16.     echo "Click on tutor name for more details<br><br>";
17.     echo "<table class='all' border='1'><tr>";
18.     echo "<th>Name</th><th>Email</th><th>Subjects to teach</th><th>Update Subjects</th><th>Delete tutor</th></tr>";
19.     foreach ($varray as $tutor) {
20.         echo "<tr>";
21.         echo "<td> <a href= view.php?id=" . $tutor->ID . ">" . $tutor->Name . "</td>";
22.         echo "<td>" . $tutor->Email . " </td>";
23.         echo "<td>" . $tutor->Subjects_to_teach . " </td>";
24.         echo "<td> <a href= update.php?id=" . $tutor->ID . "> update </td>";
25.         echo "<td> <a href= delete.php?id=" . $tutor->ID . "> delete </td>";
26.         echo "</tr>";
27.     }
28.     echo "</table>";
29.     ?>
30. </body>
31.
32. </html>

```

view.php

```
1. <html>
2.
3. <head>
4.     <title>View Our tutors</title>
5.     <link rel=stylesheet type=text/css href="st.css?v=<?php echo time();?>">
6. </head>
7.
8. <body>
9.     <?php
10.     echo "<h1>Online Academy</h1>";
11.     echo "<nav><h3>";
12.     echo "<a href='index.php'>Home</a> | <a href='Our tutors.php'>Our Tutors</a> | <a href='add.php'>Add tutor</a>";
13.     echo "</h3></nav>";
14.     include '../private/initialize.php'; // initialize the web site
15.     $id = $_GET['id'];
16.     if (!$id) {
17.         echo "problem";
18.     } else {
19.         $specific_tutor = Tutor::find_by_id($id);
20.         echo " <p> Details of the selected item </p> ";
21.
22.         echo "<table>";
23.         echo "<tr> <td><b> Tutor id </b> </td> <td>" . $specific_tutor->ID . "</td> </tr>";
24.         echo "<tr> <td> <b> Name </b> </td> <td>" . $specific_tutor->Name . "</td> </tr>";
25.         echo "<tr> <td> <b> Email </b> </td> <td>" . $specific_tutor->Email . "</td> </tr>";
26.         echo "<tr> <td> <b> Phone number </b> </td> <td>" . $specific_tutor->Phone_number . "</td> </tr>";
27.         echo "<tr> <td> <b> Qualifications </b> </td> <td>" . $specific_tutor->Qualifications . "</td> </tr>";
28.         echo "<tr> <td> <b> Subjects to teach </b> </td> <td>" . $specific_tutor->Subjects_to_teach . "</td> </tr>";
29.         echo "<tr> <td> <b> Rate per hour </b> </td> <td>" . $specific_tutor->Rate_per_hour . "</td> </tr>";
30.         echo "</table>";
31.     }
32.     ?>
33. </body>
34.
35. </html>
```

update.php

```
1. <html>
2.
3. <head>
4.     <title>Update Our tutors</title>
5.     <link rel=stylesheet type=text/css href="st.css?v=<?php echo time();?>">
6. </head>
7.
8. <body>
9.     <?php
10.     echo "<h1>Online Academy</h1>";
11.     echo "<nav><h3>";
```

```

12.     echo "<a href='index.php'>Home</a> | <a href='Our tutors.php'>Our Tutors</a> | <a href='add.php'>Add tutor</a>";
13.     echo "</h3></nav>";
14.     include '../private/initialize.php'; // initialize the web site
15.     $id = $_GET['id'];
16.     $specific_tutor = Tutor::find_by_id($id); //call the find_by_id() function; we identify the record to be updated; get all its values
17.
18.     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
19.         //if there is a post action create the array $tutorDetails and use the function merge_attributes($tutorDetails)
20.         $tutorDetails = []; //in $tutorDetails array we collect all the new values
21.         $tutorDetails['Name'] = $_POST['Name'];
22.         $tutorDetails['Email'] = $_POST['Email'];
23.         $tutorDetails['Phone_number'] = $_POST['Phone_number'];
24.         $tutorDetails['Qualifications'] = $_POST['Qualifications'];
25.         $tutorDetails['Subjects_to_teach'] = $_POST['Subjects_to_teach'];
26.         $tutorDetails['Rate_per_hour'] = $_POST['Rate_per_hour'];
27.         $specific_tutor->merge_attributes($tutorDetails);
28.         $results = $specific_tutor->update();
29.         if ($results) {
30.             echo "<br>Successful updating";
31.         }
32.         echo " <p> Details of the updated item </p> ";
33.         echo "<table>";
34.         echo "<tr> <td><b> Tutor id </b> </td> <td>" . $specific_tutor->ID . "</td> </tr>";
35.         echo "<tr> <td> <b> Name </b> </td> <td>" . $specific_tutor->Name . "</td> </tr>";
36.         echo "<tr> <td> <b> Email </b> </td> <td>" . $specific_tutor->Email . "</td> </tr>";
37.         echo "<tr> <td> <b> Phone number </b> </td> <td>" . $specific_tutor->Phone_number . "</td> </tr>";
38.         echo "<tr> <td> <b> Qualifications </b> </td> <td>" . $specific_tutor->Qualifications . "</td> </tr>";
39.         echo "<tr> <td> <b> Subjects to teach </b> </td> <td>" . $specific_tutor->Subjects_to_teach . "</td> </tr>";
40.         echo "<tr> <td> <b> Rate per hour </b> </td> <td>" . $specific_tutor->Rate_per_hour . "</td> </tr>";
41.         echo "</table>";
42.     } else {
43.         //if there is no post action present the values of the record in a form to be updated
44.         echo " <p> Use the following form to update the selected item <br> ";
45.         echo "<form action='update.php?id=" . $id . "' method='post'>";
46.         echo "<table>";
47.         echo "<tr> <td> Subjects_to_teach </td> <td> <textarea name='Subjects to teach' rows='5' cols='30'>$specific_tutor->Subjects_to_teach</textarea> </td> </tr>";
48.         echo "</table>";
49.         echo "<input type='text' name ='Name' value='$specific_tutor->Name' >";
50.         echo "<input type='text' name ='Email' value='$specific_tutor->Email' >";
51.         echo "<input type='text' name ='Phone_number' value='$specific_tutor->Phone_number' >";
52.         echo "<input type='text' name ='Qualifications' value='$specific_tutor->Qualifications' >";
53.         echo "<input type='text' name ='Rate_per_hour' value='$specific_tutor->Rate_per_hour' >";
54.         echo " <input type='submit' value='Update Record' />";
55.         echo "</form>";
56.     }

```

```

57.     ?>
58. </body>
59.
60. </html>

```

Add.php

```

1. <html>
2.
3. <head>
4.     <title>Add Our tutors</title>
5.     <link rel=stylesheet type=text/css href="st.css?v=<?php echo time();?>">
6. </head>
7.
8. <body>
9.     <?php
10.     echo "<h1>Online Academy</h1>";
11.     echo "<nav><h3>";
12.     echo "<a href='index.php'>Home</a> | <a href='Our tutors.php'>Our Tutors</a> | <a href='add.php'>Add tutor</a>";
13.     echo "</h3></nav>";
14.     include '../private/initialize.php'; // initialize the web site
15.     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
16.
17.         $tutorDetails = []; //in $tutorDetails array we
            collect all the new values
18.         $tutorDetails['Name'] = $_POST['Name'];
19.         $tutorDetails['Email'] = $_POST['Email'];
20.         $tutorDetails['Phone_number'] = $_POST['Phone_number'];
21.         $tutorDetails['Qualifications'] = $_POST['Qualifications'];
22.         $tutorDetails['Subjects_to_teach'] = $_POST['Subjects_to_teach'];
23.         $tutorDetails['Rate_per_hour'] = $_POST['Rate_per_hour'];
24.
25.         $tutor = new Tutor; // we create a new object with the values we just entered
26.         $tutor->Name = $tutorDetails["Name"];
27.         $tutor->Email = $tutorDetails["Email"];
28.         $tutor->Phone_number = $tutorDetails["Phone_number"];
29.         $tutor->Qualifications = $tutorDetails["Qualifications"];
30.         $tutor->Subjects_to_teach = $tutorDetails["Subjects_to_teach"];
31.         $tutor->Rate_per_hour = $tutorDetails["Rate_per_hour"];
32.
33.         $results = $tutor->create(); // we call the function 'create' that is a method of tutor class
34.
35.         if ($results) {
36.             echo "New Record added successfully";
37.         }
38.     } else {
39.
40.         echo " <p> Use the following form to enter details for the new item (record / object) <br> ";
41.
42.         echo "<form action=add.php method='post'>";
43.         echo "<table>";
44.         echo "<tr> <td> Name </td> <td> <input type='text' name ='Name'> </td> </tr>";
45.         echo "<tr> <td> Email </td> <td> <input type='text' name ='Email'> </td> </tr>";

```

```

46.         echo "<tr> <td> Phone number </td> <td> <input type='text' name ='Phone_number'
> </td> </tr>";
47.         echo "<tr> <td> Qualifications </td> <td> <input type='text' name ='Qualificati
ons'> </td> </tr>";
48.         echo "<tr> <td> Subjects to teach </td> <td> <textarea name='Subjects_to_teach'
rows='5' cols='30'> </textarea> </td> </tr>";
49.         echo "<tr> <td> Rate per hour </td> <td> <input type='text' name ='Rate_per_hou
r'> </td> </tr>";
50.         echo "</table>";
51.         echo " <br> <br> <input type='submit' value='Add New' />";
52.         echo "</form>";
53.     }
54.     ?>
55. </body>
56.
57. </html>

```

delete.php

```

1. <html>
2.
3. <head>
4.     <title>Delete Our tutors</title>
5.     <link rel=stylesheet type=text/css href="st.css?v=<?php echo time();?>">
6. </head>
7.
8. <body>
9.     <?php
10.         echo "<h1>Online Academy</h1>";
11.         echo "<nav><h3>";
12.         echo "<a href='index.php'>Home</a> | <a href='Our tutors.php'>Our Tutors</a> | <a h
ref='add.php'>Add tutor</a>";
13.         echo "</h3></nav>";
14.         include '../private/initialize.php'; // initialize the web site
15.         $id = $_GET['id'];
16.         $deleted_tutor = Tutor::find_by_id($id);
17.         $deleted_tutor->delete();
18.         echo "<p> The following entry is deleted </p>";
19.         echo "<table>";
20.         echo "<tr> <td> <b> Tutor id </b> </td> <td>" . $deleted_tutor-
>ID . "</td> </tr>";
21.         echo "<tr> <td> <b> Name </b> </td> <td>" . $deleted_tutor->Name . "</td> </tr>";
22.         echo "<tr> <td> <b> Email </b> </td> <td>" . $deleted_tutor-
>Email . "</td> </tr>";
23.         echo "<tr> <td> <b> Phone number </b> </td> <td>" . $deleted_tutor-
>Phone_number . "</td> </tr>";
24.         echo "<tr> <td> <b> Qualifications </b> </td> <td>" . $deleted_tutor-
>Qualifications . "</td> </tr>";
25.         echo "<tr> <td> <b> Subjects to teach </b> </td> <td>" . $deleted_tutor-
>Subjects_to_teach . "</td> </tr>";
26.         echo "<tr> <td> <b> Rate per hour </b> </td> <td>" . $deleted_tutor-
>Rate_per_hour . "</td> </tr>";
27.         echo "</table>";
28.     ?>
29. </body>
30.
31. </html>

```