Lab 06 Requirements

- Internet connectivity & VMware Workstation version 15.5.7 or above

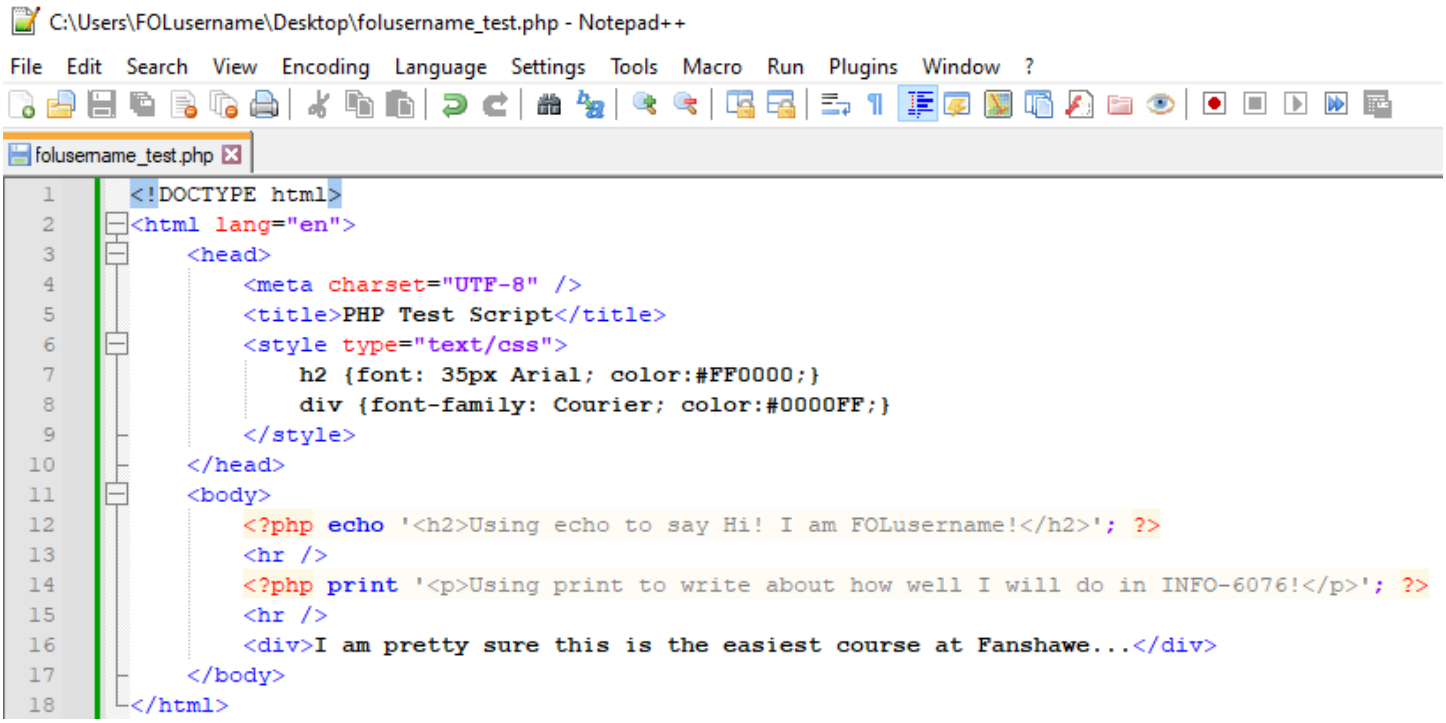
---

Part 01: Build a Basic PHP Script

On your Windows 10 VM

- Launch a text editor such as Notepad++
- Create a new file called **folusername_test.php** and save it to your desktop folder

Input the following code into the file:



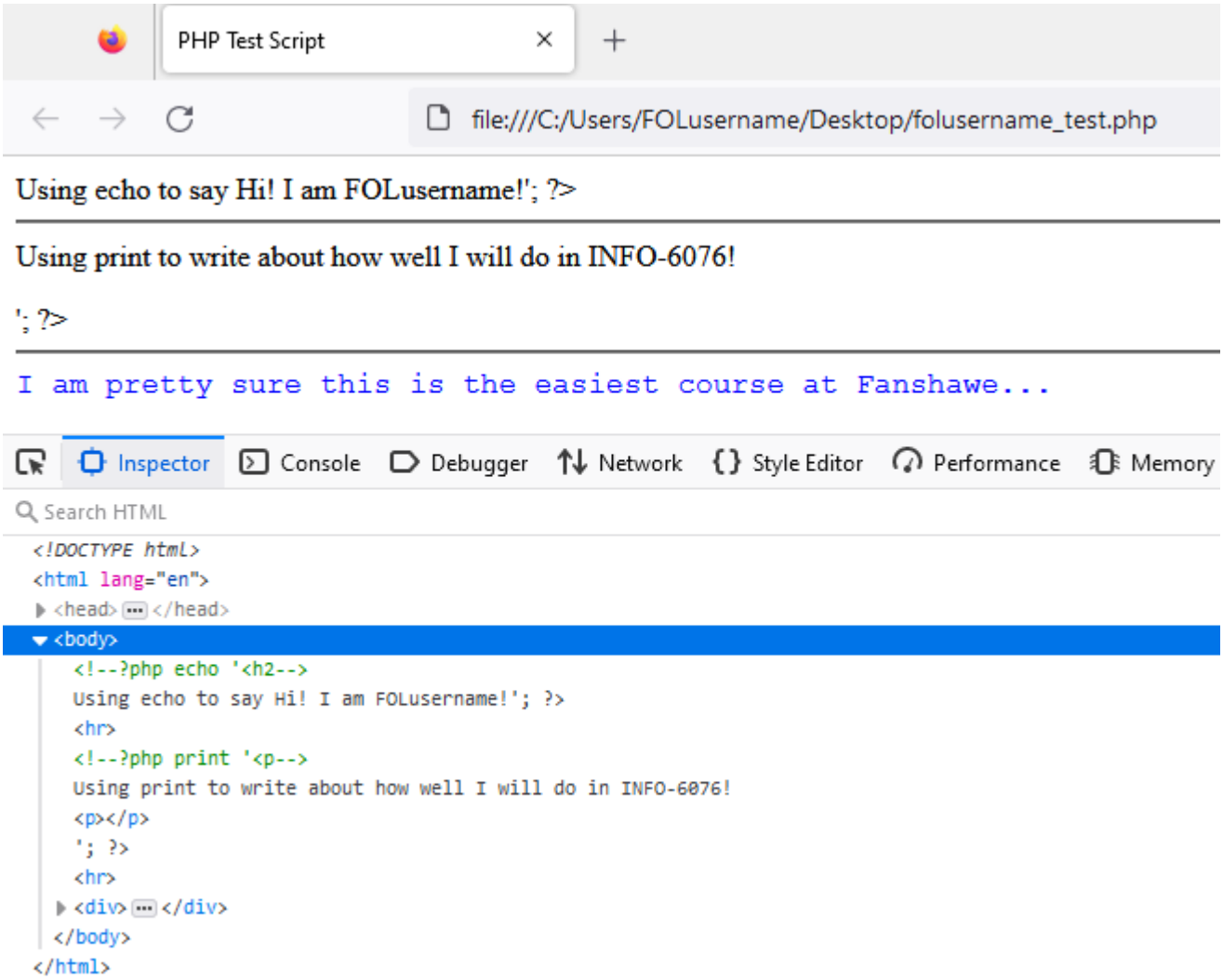

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <title>PHP Test Script</title>
        <style type="text/css">
            h2 {font: 35px Arial; color:#FF0000;}
            div {font-family: Courier; color:#0000FF;}
        </style>
    </head>
    <body>
        <?php echo '<h2>Using echo to say Hi! I am FOLusername!</h2>'; ?>
        <hr />
        <?php print '<p>Using print to write about how well I will do in INFO-6076!</p>'; ?>
        <hr />
        <div>I am pretty sure this is the easiest course at Fanshawe...</div>
    </body>
</html>
```

Save the file to your desktop folder

Open the file from your desktop using Firefox

Your browser should display an URL similar to the one below:

file:///C:/Users/FOLusername/Desktop/folusernamе_test.php



IMPORTANT!

Notice that in both cases you have **"; ?>** text on the 1st and 4th lines.

Use Firefox to inspect the code:

Take your mouse coursor to Using echo
Right mouse click -> select Inspect Element (Q)

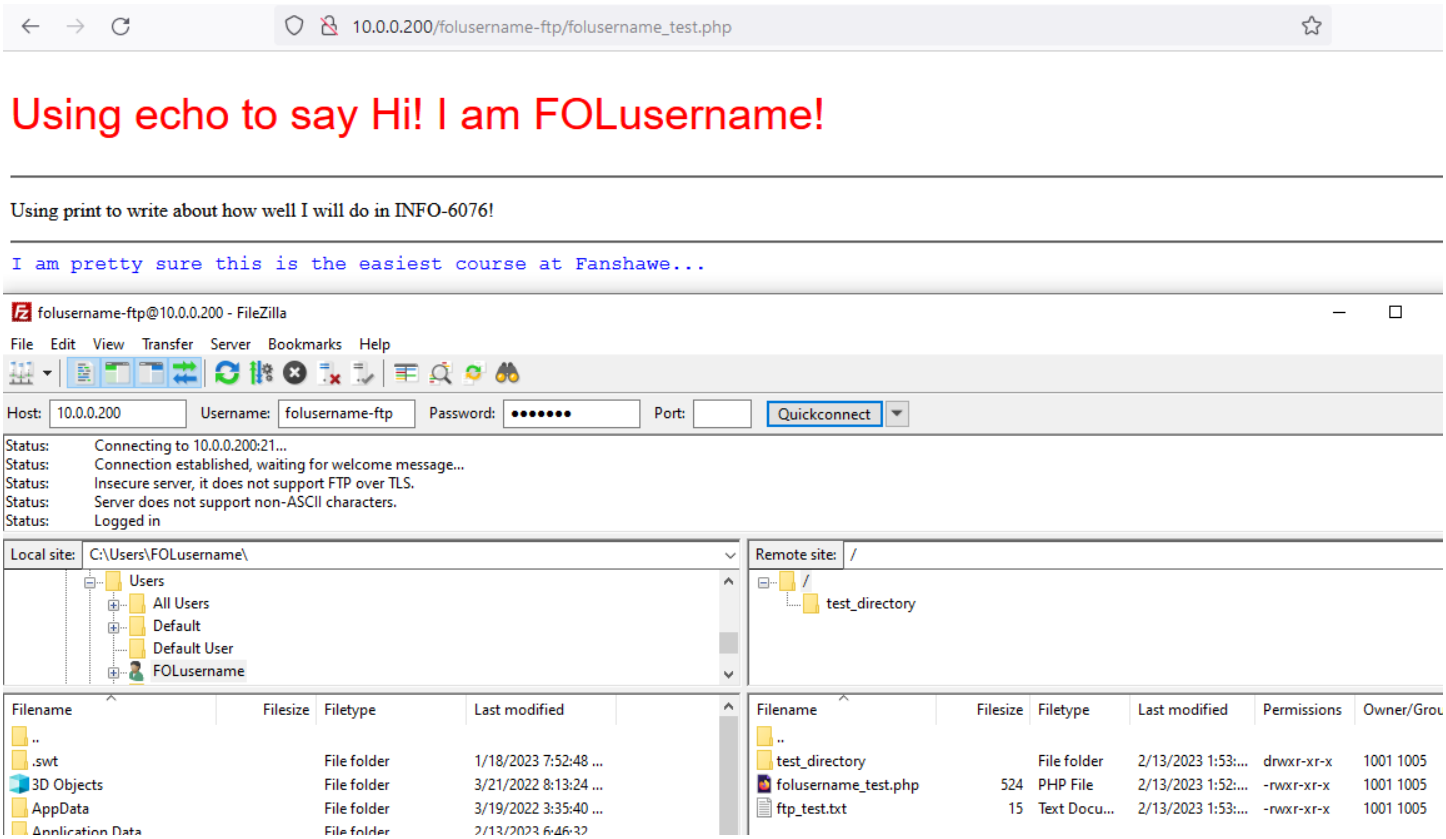
Did you notice the PHP code that is commented out by the Firefox browser?

Modern browsers can recognize unprocessed PHP code inside HTML pages and usually comment out any PHP functions

The page **folusername_test.php** wasn't processed by a PHP Interpreter because we don't have PHP installed on our Windows 10 VM

Use FileZilla to upload the file to your Ubuntu Web Server using the username/password you created in Lab 03 for FTP transfers. Verify that your uploaded file is accessible through the browser on W10

Open the file **folusername_test.php** in FireFox by navigating to the appropriate URL for your Ubuntu Web Server:



Slide 01:

- Take a screenshot showing the page being served by the Ubuntu Web Server
- Show the successful upload through FTP and include your FOLusername

Add variables, date and printf() function to your PHP script

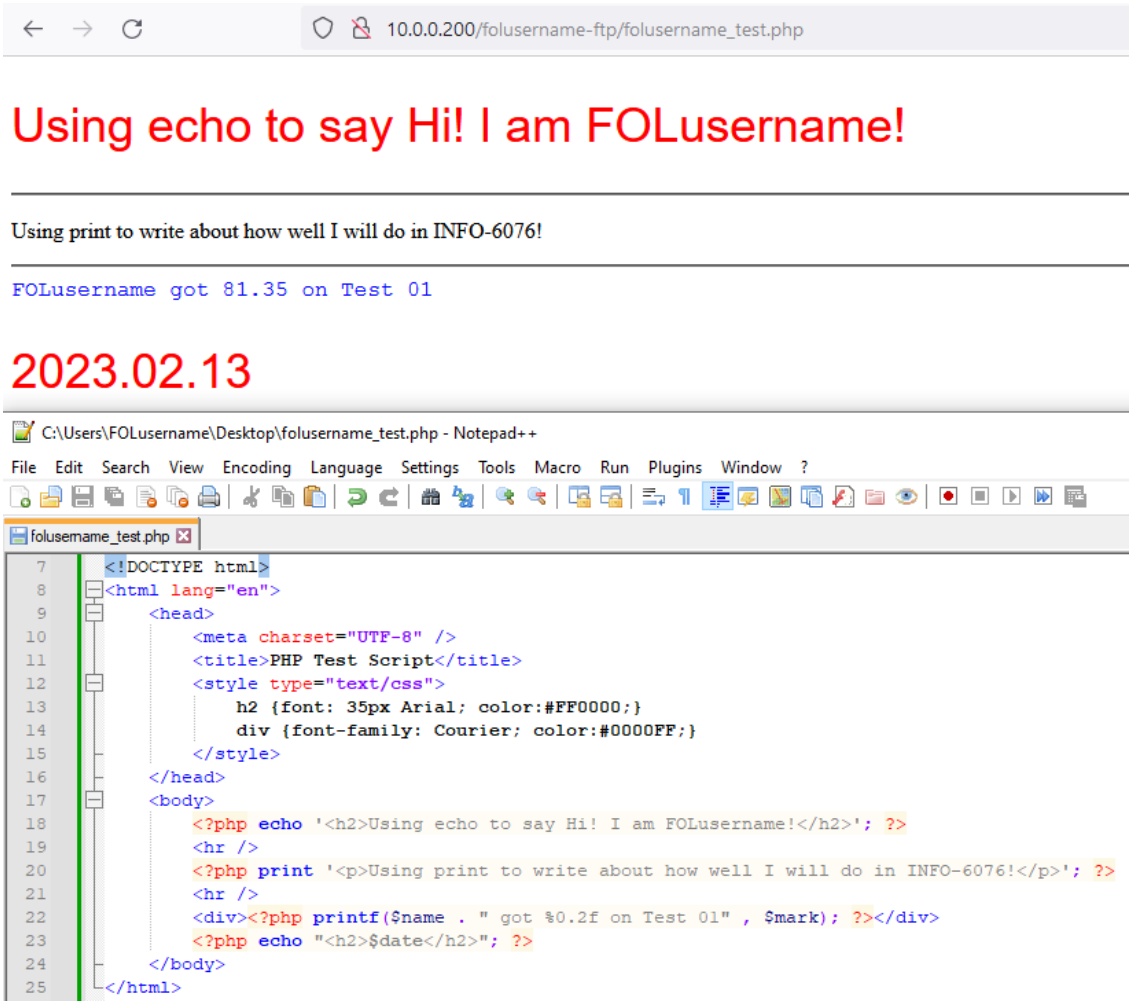
- Add the following block of code before <!doctype html>:

```
<?php
    $mark=81.3456;
    $date= date('Y.m.d');
    $name= "FOLusername";
 ?>
```

- Replace the text inside the tag (between and) with:
  <?php printf($name . " got %0.2f on Test 01." , $mark); ?>

- Add the following line after the tag and before the </body> tag:
  <?php echo "<h2>$date</h2>"; ?>

Save the file and upload it to the Web Server. Then refresh the window in FireFox:



Slide 02:
Take a screenshot showing all of the above and place it into slide 02

Finally, we can use a PHP function to access files on the server

In this Lab we are going to include the hidden **accounts.txt** file with Mutillidae passwords (recall Lab-04 Part 04).

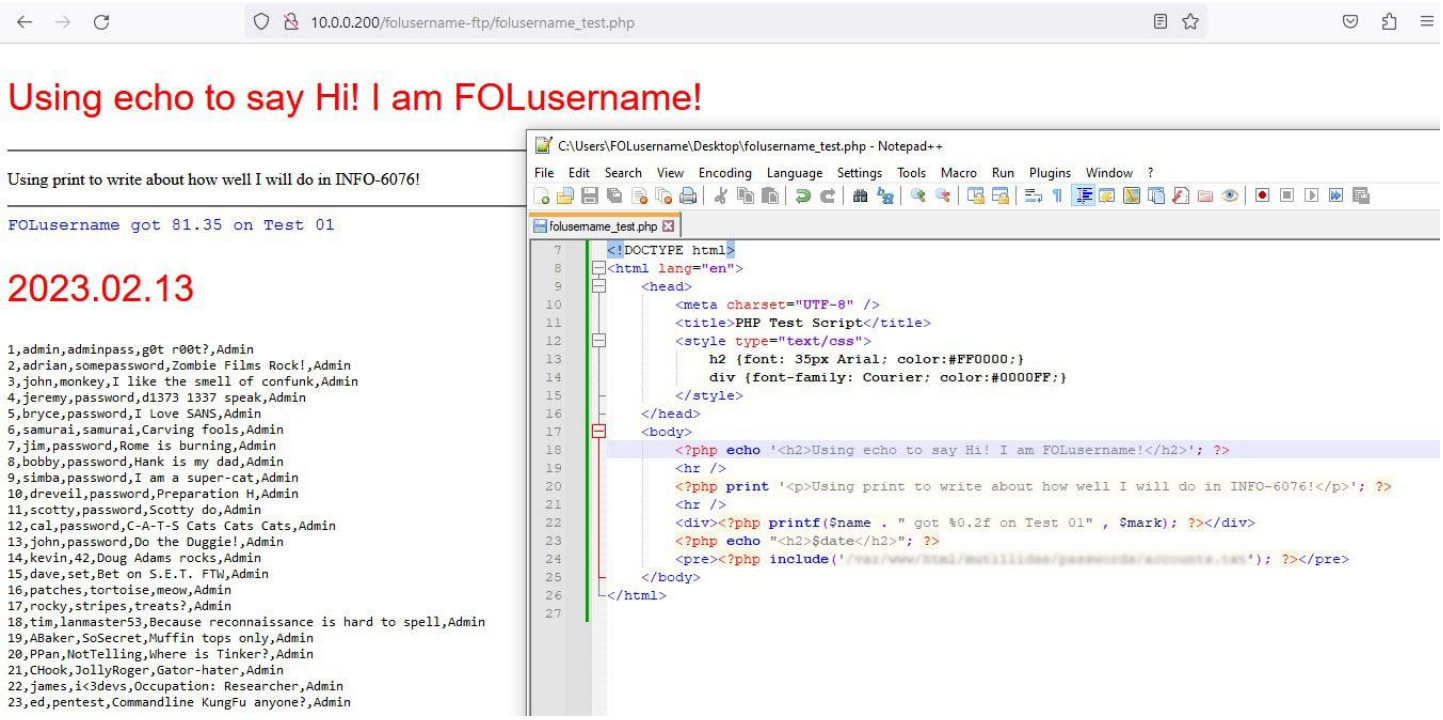
You will need to navigate the directories on the Ubuntu Web Server in order to get the desired information

<u>Hint:</u>

Before the ending </body>) tag add a line using the PHP include function

<pre><?php include(' ******** '); ?></pre>**

Save the file and upload it to the web server then refresh the window in FireFox:



Slide 03:

- Take a screenshot showing the page with **accounts.txt** included
- Show the PHP include statement you used

Part 02: Working with SQL databases

We are now going to take a look at how to create a database using MySQL

On your Ubuntu Server, open the terminal and switch to the super user:

sudo su

Open MySQL monitor and login:

mysql –u root -p

You will be prompted for a password. It should be blank (no password)

View the existing databases with the show databases command:

SHOW databases;

Create a database with a couple sample tables to test our privileges

CREATE database FOLusername;

Work with the database you just created by using the USE command:

USE FOLusername;

Create two tables with the following commands:

CREATE table Test1 (ID tinyint(2));
CREATE table Test2 (ID tinyint(2), Name varchar(200));

List all the tables in the **FOLusername** database:

SHOW tables;

Check table structure with the following commands (**DESC** is a shortcut for **DESCRIBE**):

DESC Test1;
DESC Test2;

Remember that table names are CaSe_SenSiTivE

Create New Users and Test Privileges

Create a new user **FOLusername** with the following commands.
Note: For clarity SQL commands are on separate lines. You need to hit enter at the end of each line.

Note: If you grant privileges to a user that does not exist, MySQL will create the user.

GRANT ALL
ON *.*
TO FOLusername@localhost
IDENTIFIED BY 'Windows1';

Use the **exit** command to exit from MySQL monitor.

Log in to MySQL as a different User

Log back on with the **FOLusername** user you just created:

mysql -u FOLusername -p

The **-p** option prompts you to enter a password, if you don't add this option the command will error out.

Use the **SHOW databases;** command to view the list of databases available to the new user

Setting a user's permissions for SQL

Exit and log back on as user **root**.

Create a new user **FOLusername_01** that has SELECT permissions for the **FOLusername** database and all the tables within using the following command.

GRANT SELECT
ON FOLusername.*
TO FOLusername_01@localhost
IDENTIFIED BY 'Windows1';

Exit and log back on as user **FOLusername_01**

Issue **SHOW tables** for the **FOLusername** database.

Hint: you need select (use) the **FOLusername** database to issue the command:

USE FOLusername;
SHOW tables;

Use the **status** command to confirm you are logged on as the right user

```
Database changed
mysql> show tables;
+-----------------------+
| Tables_in_FOLusername |
+-----------------------+
| Test1                 |
| Test2                 |
+-----------------------+
2 rows in set (0.00 sec)

mysql> status
--------------
mysql  Ver 14.14 Distrib 5.7.23, for Linux (x86_64) using  EditLine wrapper

Connection id:          7
Current database:       FOLusername
Current user:           FOLusername01@localhost
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.7.23-0ubuntu0.18.04.1 (Ubuntu)
Protocol version:       10
Connection:             Localhost via UNIX socket
Server characterset:    latin1
Db     characterset:    latin1
Client characterset:    utf8
Conn.  characterset:    utf8
UNIX socket:            /var/run/mysqld/mysqld.sock
Uptime:                 57 min 34 sec

Threads: 1  Questions: 34  Slow queries: 0  Opens: 115  Flush tables: 1  Open tables: 106  Queries p
er second avg: 0.009
--------------

mysql>
```

Slide 04:

- Take a screenshot showing the above and place it into slide 04

Limiting User's permissions in MySQL

Exit and log back on as user **root**

Create a new user **FOLusername**_02 that has SELECT permissions only for the Test2 table within the FOLusername database with a password of **Windows1**

GRANT SELECT
ON FOLusername.Test2
TO FOLusername_02@localhost
IDENTIFIED BY 'Windows1';

Exit and log back on as user **FOLusername_02**

Issue a **SHOW tables** for the **FOLusername** database and the mysql **status** command

Exit from the MySQL Monitor

Slide 05:

- Show the output of **show tables** and the **status** commands in SQL
- Issue the date command once back in the terminal

*** Take a snapshot of all the VMs named **After Lab 06** ***