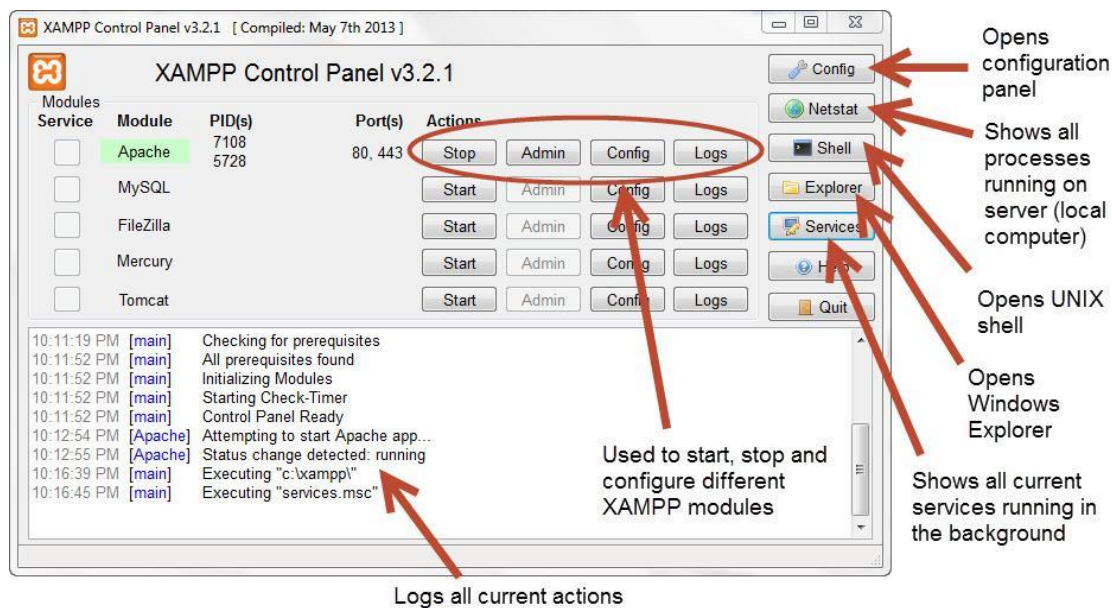


Laboratory Exercise #1 – Install PHP:

1. Go to <https://www.apachefriends.org/index.html>
2. Download latest XAMPP for your OS
3. Install XAMPP by clicking on the application and follow instructions
4. Once installed open up your XAMPP application, you should see something like this:



5. Start your XAMPP installation and go to <http://127.0.0.1> or <http://localhost>
6. If you see something like this below – your install was successful.



Installing LAMP on Linux

For our purposes, we will be installing LAMP on Ubuntu (VM provided).

1. Run the VM on VMWare Workstation. The Login and Password is php7 / P@ssw0rd
2. Update the repository cache by running:
`sudo apt-get update`
3. Install Apache and press y to install:
`sudo apt-get install apache2`
4. To check if apache installed correctly, run:
`sudo service apache2 status`
5. If the VM has UFW Firewall installed, we need to allow traffic on ports 80 and 443. Run the following command:
`sudo ufw app list`
`sudo ufw app info "Apache Full"`
6. Let us install MySQL – we will learn how to do basic administration on the laboratory exercises later.
`sudo apt-get install mysql-server`
7. Lets install php (for ubuntu, this is php7.2) and common extensions for our use later on:
`sudo apt-get install php7.2 php7.2-common php-pear php7.2-gd php7.2-mbstring php7.2-zip php7.2-mysql php7.2-xml`
8. Lets create a test file:
`cd /var/www/html`
`nano info.php`
9. In info.php, enter the following:
`<?php phpinfo(); ?>`
10. Save the file then restart apache:
`sudo systemctl restart apache2`
11. Get the IP address of the VM:
`ifconfig`
12. Browse to the IP address on your browser using the url:
`<ip address>/info.php`

Laboratory Exercise #2 – Hello World

1. Open up your favorite text editor and type the following:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
<title>Hello, World!</title>
```

```

</head>
<body>
<?php
echo "Hello, World!";
// You could have used the print statement here.
?>
</body>
</html>

```

2. Save your text as test.php in c:\xampp\htdocs\ (or where your XAMPP install was placed)
3. Browse to <http://localhost/test.php>
4. If you see a "Hello, World!" output, you are successful in creating a hello world program.

Laboratory Exercise #3 – Create and Use Variables

1. echo the following statement to the browser: "Twinkle, Twinkle little star."
2. Next, create two variables, one for the word "Twinkle" and one for the word "star".
3. Echo the statement to the browser, this time substituting the variables for the relevant words.
4. Change the value of each variable to whatever you like, and echo the statement a third time. Remember to include code to show your statements on different lines.

Laboratory Exercise #4 – Arithmetic Functions

1. Create the following variables: \$x = 10 and \$y = 7.
2. Write code to print out the following, while saving the temporary result in a variable named \$result:

```

10 + 7 = 17
10 - 7 = 3
10 * 7 = 70
10 / 7 = 1.4285714285714
10 % 7 = 3

```

Laboratory Exercise #5 – Arithmetic Assignment Operators and Variables

Arithmetic-assignment operators perform an arithmetic operation on the variable at the same time as assigning a new value.

For example:

```

$x += <num> adds a value number to the variable $x.
$x -= <num> subtracts a value to the variable
$x *= <num>

```

`$x /= <num>`

For increment and decrement use the `$x++` and `$x--` functions accordingly.

For the exercise below, reproduce the following output:

Use the format "Value is now \$var"

Value is now 8.

Add 2. Value is now 10.

Subtract 4. Value is now 6.

Multiply by 5. Value is now 30.

Divide by 3. Value is now 10.

Increment value by one. Value is now 11.

Decrement value by one. Value is now 10.

Laboratory Exercise #6 – Showing what is inside the variables

1. The functions `print_r` and `var_dump` show the kind and the value of a specified variable. Try looking for these variables at php.net
2. For this exercise try using these functions to output exactly what happens below:

```
string(5) "Teddy"  
Teddy  
int(45)  
NULL
```

Laboratory Exercise #7 – Concatenation operator

The "." Operator concatenates two strings. For this exercise, assuming a variable named `$around` with the value "around", use the concatenation operator to output the text:

What goes around comes around.

Laboratory Exercise #8 – The `gettype()` function

1. Look up the `gettype` function in the php.net online manual. What does it say?
2. With the format "Value is <type>", output the following to the screen (using appropriate line breaks):

```
Value is string.  
Value is double.  
Value is boolean.  
Value is integer.
```

Value is NULL.

3. Make sure that the type is stored in a variable of your choosing.

Additional Laboratory Exercises:

Decision Making

1. Write a program CheckOkFail which prints "OK" if the int variable "testVar" is more or equal to 75, otherwise, it prints "Fail".
2. Write a program named OddOrEven which prints "Odd" if the int variable "number" is an odd number or "Even" if the number is even. Hint: use the % operator
3. Write a program named NumberToWord which prints the equivalent word number "One", "Two",... for the corresponding int "number" value of 1, 2, 3... This uses the switch statement.