SAVITRIBAI PHULE PUNE UNIVERSITY

LAB COURSE III

INTERNET PROGRAMMING, NETWORKING & PROJECT

(COURSE CODE:CS-348)

T.Y.B.SC.(COMPUTER SCIENCE)

SEMESTER - I

Name ____

College Name ____

Roll No. ______________________ Division _____________________

Academic Year ____________

Internal Examiner : ---------------------External Examiner : ---------------------

PREPARED BY

MRS. SHILPA KHADILKAR (KAVERI COLLEGE)
MRS. VANDANA BARBEKAR (WADIA COLLEGE)
MRS. SANGEETA RAUT (MODERN COLLEGE)

FIRST EDITION AUTHORS:

SHINDE SAHEBRAO
NIKAM PRADEEP
JOSHI GAJANAN
MAHAJAN NILESH
MUSALE VAIBHAV
SATPUTE JAYA
TANK BHUPESH
MANKAR ABHIJEET
JOSHI VARSHA
MAHAJAN MANISHA
RAYATE RUPALI

BOARD OF STUDY (COMPUTER SCIENCE) MEMBERS:

Dr. Shailaja C. Shirwaikar

About The Work Book

Objectives –

1. The scope of the course.

2. Bringing uniformity in the way course is conducted across different Colleges.

3. Continuous assessment of the students.

4. Providing ready references for students while working in the lab.

How to use this book?

This book is mandatory for the completion of the laboratory course. It is a

Measure of the performance of the student in the laboratory for the entire duration of the course.

Instructions to the students

1) Students should carry this book during practical sessions of Computer Science.

2) Students should maintain separate journal for the source code and outputs.

3) Students should read the topics mentioned in reading section of this Book before coming for practical.

4) Students should solve all exercises which are selected by Practical in-charge as a part of journal activity.

5) Students will be assessed for each exercise on a scale of 5

1	Note done	0
2	Incomplete	1
3	Late complete	2
4	Needs improvement	3
5	Complete	4
6	Well-done	5

Instructions to the practical in-charge

1. Explain the assignment and related concepts in around ten minutes using white board if required or by demonstrating the software.
2. Choose appropriate problems to be solved by student.
3. After a student completes a specific set, the instructor has to verify the outputs and sign in the provided space after the activity.
4. Ensure that the students use good programming practices.
5. You should evaluate each assignment carried out by a student on a scale of 5 as specified above ticking appropriate box.
6. The value should also be entered on assignment completion page of respected lab course.

PHP Semester -1
Assignment Completion Sheet

Sr. No.	Assignment Name	Marks (out of 5)	Sign
1	To study functions & strings		
2	To study Arrays		
3	To study Files and Directories		
4	Object Oriented Programming		
5	PHP-DATABASE(PostgreSQL)		
	Total out of 25		
	Total out of 05		

ASSIGNMENT NO. 1 : TO STUDY FUNCTIONS & STRINGS

User-defined functions

A function may be defined using syntax such as the following:

```
function function_name([argument_list…])
{
      [statements]
      [return return_value;]
}
```

Any valid PHP code may appear inside a function, even other functions and class definitions. The variables you use inside a function are, by default, not visible outside that function. In PHP3 functions must be defined, before they are referenced. No such requirement exists in PHP4.

Example 1.

Code	Output

<?php
msg("Hello");   // calling a function
function msg($a)   // defining a function
{
 echo $a;
}
?>
	Hello

Default parameters

You can give default values to more than one argument, but once you start assigning default values, you have to give them to all arguments that follow as well.

Example 2.

Code	Output

<?php
function display($greeting, $message=”Good Day”)
{
   echo $greeting;
   echo “
”;
  echo $message;
}
display(“Hello”);
?>
	Hello
	Good Day

Variable parameters

You can set up functions that can take a variable number of arguments. Variable number of arguments can be handled with these functions:

func_num_args : Returns the number of arguments passed
func_get_arg   : Returns a single argument
func_get_args   : Returns all arguments in an array

Example 3.

Code	Output

<?php
echo “Passing 3 arg. to xconcat
”;
echo “Result is …”;
xconcat(“How”,”are”,”you”);
function xconcat( )
{
  $ans = “”;
  $arg = func_get_args( );
  for ($i=0; $i<func_num_args( ); $i++)
  {
    $ans **.**= $arg[$i]**.**” “;
   }
    echo $ans;
}
?>
	Passing 3 arg. to xconcat
	Result is …How are you

Missing parameters

When using default arguments, any defaults should be on the right side of any non-default arguments, otherwise, things will not work as expected.

Example 4.

Code	***Output***

<?php
function makecoffee ($type = "Nescafe")
{
return "Making a cup of $type
";
}
echo makecoffee ();
echo makecoffee ("espresso");
?>
	Making a cup of Nescafe.
	Making a cup of espresso.

<?php
function make ($type = "acidophilus", $flavour)
{
return "Making a bowl of $type $flavour
";
}
echo make ("raspberry");   // won’t work
?>
	Warning: Missing argument 2 in call to make()…..
	Making a bowl of raspberry

<?php
function make ($flavour, $type = "acidophilus")
{
return "Making a bowl of $type $flavour
";
}
echo make ("raspberry");   //it works
?>
	Making a bowl of acidophilus raspberry.

Variable functions

Assign a variable the name of a function, and then treat that variable as though it is the name of a function.

Example 5.

Code	Output

<?php
$varfun='fun1';
$varfun( );
$varfun='fun2';
$varfun( );
$varfun='fun3';
$varfun( );
function fun1( )
{
 echo "
Function one";
}
function fun2( )
{
 echo "
Function two";
}
function fun3( )
{
 echo "
Function three";
}
?>
	Function one
	Function two
	Function three

Anonymous functions

The function that does not possess any name are called anonymous functions. Such functions are created using *create_function( )* built-in function. Anonymous functions are also called as lambda functions.

Example 6.

Code	Output
<?php $fname=create_function('$a,$b', '$c = $a + $b; return $c;'); echo $fname(10,20); ?>	30

Strings

Strings in PHP

- Single quoted string (few escape characters supported, variable interpolation not possible)
- Double quoted string (many escape characters supported, variable interpolation possible)
- Heredoc

There are functions to print the string, namely print, printf, echo.
The print statement can print only single value, whereas echo and printf can print multiple values. Printf requires format specifiers. If echo statement is used like a function, then only one value can be printed.

Comparing Strings

Example 1.

Code	Output

<?php
$a='amit';
$b='anil';
if($a==$b)   //using operator
  echo "Both strings are equal
";
else
  echo "Both strings are not equal
";
if(strcmp($a,$b)>0)   //using function
{
 echo "String2 sorts before String1";
}
elseif(strcmp($a,$b)==0)
{
 echo "both are equal";
}
elseif(strcmp($a,$b)<0)   // negative value
{
 echo "String1 sorts before String2";
}?>
	Both strings are not equal
	String1 sorts before String2

<?php
$a=34;
$b='34';
if($a===$b)   //using operator
  echo "Both strings are equal
";
else
  echo "Both strings are not equal
";
?>
	Both strings are not equal

Other string comparison functions
strcasecmp( ) : case in-sensitive string comparison
strnatcmp( ) : string comparison using a “natural order” algorithm
strnatcasecmp( ) : case in-sensitive version of strnatcmp( )

String manipulation & searching string

Example 2.

Code	Output
<?php $small="India";	is my

```php
$big="India is my country";
$str=substr($big,6,5);
echo "<br>$str";
$cnt = substr_count($big,"i");
echo "<br>There are".$cnt." i's in $big";
$pos=strpos($big,"is");
echo "<br>is found at $pos position";
$replace=substr_replace($big,"Bharat",0,5);
echo "<br>before replacement->$big";
echo "<br>after replacement ->$replace";
?>
```

```
There are 2 i's in India is my
country

is found at 6 position

before replacement->India is
my country
after replacement ->Bharat is
my country
```

Regular Expressions
Two types of regular expressions
POSIX – style
PERL – compatible
Purpose of using regular expressions
Matching
Substituting
Splitting
Example 3.

Code

```php
<?php
$big=<<< paragraph
India is my country.
I am proud of it.
I live in Maharashtra.
paragraph;
echo "<br>";
$found=preg_match('/am/i',$big);
if($found)
  echo "<br>am found in \$big";
$replace=preg_replace('/India/','Bharat',$big);
echo "<br>$replace";
$split=preg_split('/ /',$big);
foreach($split as $elem)
{ echo "<br>$elem";}
?>
```

Output

```
am found in $big
Bharat is my country. I am
proud of it. I live in
Maharashtra.
India
is
my
country.
I
am
proud
of
it.
I
live
in
Maharashtra
```

<u>Set A</u>

Q: 1) Write a PHP script for the following: Design a form to accept a string. Write a function to count the total number of vowels (a,e,i,o,u) from the string. Show the occurrences of each vowel from the string. Check whether the given string is a palindrome or not, without using built-in function. (Use radio buttons and the concept of function. Use ‘include’ construct or require stmt.)

Q: 2) Write a PHP script for the following: Design a form to accept two strings from the user. Find the first occurrence and the last occurrence of the small string in the large string. Also count the total number of occurrences of small string in the large string. Provide a text box to accept a string, which will replace the small string in the large string. (Use built-in functions)

<u>Set B</u>

Q: 1) Write a PHP script for the following: Design a form to accept two numbers from the user. Give options to choose the arithmetic operation (use radio buttons). Display the result on the next form. (Use the concept of function and default parameters. Use 'include' construct or require stmt)

Q: 2) Write a PHP script for the following: Design a form to accept two strings from the user. Find whether the small string appears at the start of the large string. Provide a text box to accept the string that will replace all occurrences of small string present in the large string. Also split the large string into separate words. (Use regular expressions)

<u>Set C</u>

Q: 1) Write a PHP script for the following: Design a form to accept the details of 5 different items, such as item code, item name, units sold, rate. Display the bill in the tabular format. Use only 4 text boxes. (Hint : Use of explode function.)

Q: 2) Write a PHP script for the following: Design a form to accept two strings. Compare the two strings using both methods (= = operator & strcmp function). Append second string to the first string. Accept the position from the user; from where the characters from the first string are reversed. (Use radio buttons)

3. Using regular expressions check for the validity of entered email-id. The @ symbol should not appear more than once. The dot (.) can appear at the most once before @ and at the most twice or at least once after @ symbol. The substring before @ should not begin with a digit or underscore or dot or @ or any other special character. (Use explode and ereg function.)

Signature of the instructor :____________________ Date :_____________

Assignment Evaluation

0:Not Done □ 2:Late Complete □ 4:Complete □

1:Incomplete □ 3:Needs Improvement □ 5:Well Done □

ASSIGNMENT NO. 2 : TO STUDY ARRAYS

ARRAYS : An array is a collection of data values. Array is organized as an ordered collection of (key,value) pairs.

In PHP there are two kinds of arrays :

Indexed array : An array with a numeric index starting with 0.
For example,
Initializing an indexed array,
$numbers[0]=100;
$numbers[1]=200;
$numbers[2]=300;

Associative array : An array which have strings as keys which are used to access the values.
Initializing an Associative array,
$numbers[‘one’]=100;
$numbers[‘two’]=200;
$numbers[‘three’]=300;

Functions used with array :

Name	Use	Example
array()	This construct is used to initialize an array.	$numbers=array(100,200,300); $cities=array(‘Capital of Nation‘=>’Delhi’, ’Capital of state’=>’Mumbai’, ’My city’=>’Nashik’);
list()	This function is used to copy values from array to the variables.	$cities=array(‘Capital of Nation‘=>’Delhi’, ’Capital of state’=>’Mumbai’, ’My city’=>’Nashik’); List($cn,$cs,$c)=$cities; Output : $cn=’Delhi’ $cs=’Mumbai’ $c=’Nashik’
array_splice()	This function is used to remove or insert elements in array	$student=array(11,12,13,14,15,16); $new_student=array_splice($student,2,3); /* starting from index(2) and length =3 $new_student1=array_splice($student,2); /* here length is not mentioned */ Output : $new_student=(13,14,15); $new_student1=(13,14,15,16);

array_key_exists()	This function is used to check if an element exist in the array.	$cities=array(‘Capital of Nation‘=>’Delhi’, ’Capital of state’=>’Mumbai’, ’My city’=>’Nashik’);
		If (array_key_exists(‘Capital of State’,$cities))
		{
		Echo “key found!\n”;
		}

		Output : Key_found!
extract()	This function automatically creates local variables from the array.	Extract($student); By this, the variables are created like this : $roll = 11; $name=’A’; $class=’TYBSc’;

foreach()	This is the most common way to loop over elements of an array.
	PHP executes the body of the loop once for each element of $students, with $value set to the current element.
		For indexed array :
		
		$students=array(‘a’,’b’,’c’,’d’);
		Foreach($student as $value)
		{
		Echo “student $value \n”;
		}
		Output
		Student A
		Student B
		Student C
		Student D
		For associative array :
		$students=array(‘Name’=>‘a’,’Roll no’ => 100, ‘class’=>’TYBSc’);
		Foreach($student as $key=>$value)
		{
		Echo “student’s $key is : $value \n”;
		}
		
		Student’s Name is : A
		Student’s Roll No is : 100
		Student’s class is : TYBSC

array_push()
array_pop()
	These functions are used to treat an array like a stack .	Array_push(a);
		Array_pop(a);

array_shift()
array_unshift()
	These functions are used to treat an array like a queue.	Array_shift();
		Array_unshift();

Set A

Q: 1) Write a menu driven program to perform the following operations on an associative array:

a) Display the elements of an array along with the keys.
b) Display the size of an array
c) Delete an element from an array from the given key/index.
d) Reverse the order of each element’s key-value pair [Hint: use array_flip()]
e) Traverse the elements in an array in random order [[Hint: use shuffle()].

Q:2) Accept a string from the user and check whether it is a palindrome or not (Implement stack operations using array built-in functions).

Set B

Q: 1) Declare a Multidimensional Array. Display specific element from a Multidimensional array. Also delete given element from the Multidimensional array.(After each operation display array content [Hint : use print_r()])

Q: 2) Define an array. Find the elements from the array that matches the given value using appropriate search function.

Set C

Q: 1) Write a menu driven program to perform the following stack and queue related operations:[Hint: use Array_push(), Array_pop(), Array_shift(), Array_unshift()]

a) Insert an element in stack
b) Delete an element from stack
c) Display the contents of stack
d) Insert an element in queue
e) Delete an element from queue
f) Display the contents of queue

Q: 2) Write a menu driven program to perform the following operations on associative arrays:

a) Sort the array by values (changing the keys) in ascending, descending order.
b) Also sort the array by values without changing the keys.
c) Filter the odd elements from an array.
d) Sort the different arrays at a glance using single function.
e) Merge the given arrays.
f) Find the intersection of two arrays.
g) Find the union of two arrays.
h) Find set difference of two arrays.

Signature of the instructor :____________________ Date :_____________

Assignment Evaluation

0:Not Done	□	2:Late Complete	□	4:Complete	□
1:Incomplete	□	3:Needs Improvement	□	5:Well Done	□

ASSIGNMENT NO. 3 : TO STUDY FILES AND DIRECTORIES

File : A **file** is nothing more than an ordered sequence of bytes stored on hard disk, floppy disk CD-ROM or some other storage media. Operations on file are
Opening and closing a file.
Reading a file and writing into file
Deleting and renaming a file
Navigating a file
Opening and closing directories
Reading directory entries
Deleting and renaming a directory
Note:- one differences between Linux and windows when it comes to specifying directory path is UNIX based system like LINUX use forward slash to delimit elements in a path
A **file handle** is nothing more than an integer value that will be used to identify the file you wish to work with until it is closed working with files

<table>
<tr><th>Function Name</th><th>Description</th></tr>
<tr><td>fopen()</td><td>Opening and closing a file
This is used to open a file ,returning a file handle associated with opened file .It can take three arguments :fname,mode and optional use_include_path
Ex:-$fp=fopen(“data.txt”,r);
We can also open a file on remote host
List of modes used in fopen are:
<table>
<tr><th>Mode</th><th>Purpose</th></tr>
<tr><td>R</td><td>Open for reading only; place the file pointer at the beginning of the file</td></tr>
<tr><td>r+</td><td>Open for reading and writing; place the file pointer at the beginning of the file.</td></tr>
<tr><td>w</td><td>Open for writing only; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.</td></tr>
<tr><td>w+</td><td>Open for reading and writing; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.</td></tr>
<tr><td>A</td><td>Open for writing only; place the file pointer at the end of the file. If the file does not exist, attempt to create it.</td></tr>
<tr><td>a+</td><td>Open for reading and writing; place the file pointer at the end of the file. If the file does not exist, attempt to create it.</td></tr>
</table>
</td></tr>
<tr><td>fclose()</td><td>This is used to close file, using its associated file handle as a</td></tr>
</table>

	single argument Ex:- fclose(fp);
fread( )	This function is used to extract a character string from a file and takes two arguments, a file handle and a integer length Ex: fread($fp,10);
fwrite()	This function is used to write data to a file and takes two arguments, a file handle and a string Ex: fwrite($fp,"HELLO");
fgetc()	Function can be used to read one character from file at a fileIt takes a single argument ,a file handle and return just one character from the file .It returns false when it reached to end of file.
fgets()	This function is used to read set of characters it takes two arguments, file pointer and length. It will stop reading for any one of three reasons: The specified number of bytes has been read A new line is encountered The end of file is reached
fputs()	This is simply an alias for fwrite() .
file()	This function will return entire contents of file.This function will automatically opens,reads,anclose the file.It has one argument :a string containing the name of the file.It can also fetch files on remote host.
fpassthru()	This function reads and print the entire file to the web browser.This function takes one argument ,file handle.If you read a couple of lines from a file before calling fpassthru() ,then this function only print the subsequent contents of a file.
readfile()	This function prints content of file without having a call to fopen() It takes a filename as its argument ,reads a file and then write it to standard output returning the number of bytesread(or false upon error)
fseek()	It takes file handle and integer offset , offset type as an arguments .It will move file position indicator associated with file pointer to a position determined by offset. By default this offset is measured in bytes from the beginning of the file. The third argument is optional ,can be specified as: SEEK_SET:-Beginning of file +offset SEEK_CUR:-Current position +offset(default) SEEK_END:-End of the file +offset
ftell()	It takes file handle as an argument and returns the current offset(in bytes) of the corresponding file position indicator.
rewind()	It accepts a file handle as an argument and reset the corresponding file position indicator to the beginning of file.
file_exists()	It takes file name with detail path as an argument and returns true if file is there otherwise it returns false
file_size()	It takes file name as an argument and returns total size of file

<table>
<tr><td></td><td>(in bytes)</td></tr>
<tr><td>fileatime()</td><td>It takes filename as an argument and returns last access time for a file in a UNIX timestamp format</td></tr>
<tr><td>filectime()</td><td>It takes filename as an argument and returns the time at which the file was last changed as a UNIX timestamp format</td></tr>
<tr><td>filemtime()</td><td>It takes filename as an argument and returns the time at which the file was last modified as a UNIX timestamp format</td></tr>
<tr><td>fileowner()</td><td>It takes filename as an argument and returns the user ID of the owner of specified file.</td></tr>
<tr><td>posix_getpwuid()</td><td>It accept user id returned by fileowner() function as an argument and returns an associative array with following references
<table>
<tr><td>Key</td><td>Description</td></tr>
<tr><td>name</td><td>The shell account user name of the user</td></tr>
<tr><td>passwd</td><td>The encrypted user password</td></tr>
<tr><td>Uid</td><td>The ID number of the user</td></tr>
<tr><td>Gid</td><td>The group ID of the user</td></tr>
<tr><td>Gecos</td><td>A comma separated list containing user full name office phone, office number and home phone number</td></tr>
<tr><td>Dir</td><td>The absolute path to the home directory of the user</td></tr>
<tr><td>Shell</td><td>The absolute path to the users default shell</td></tr>
</table></td></tr>
<tr><td>filegroup()</td><td>It takes filename as an argument and returns the group ID of owner of the specified file</td></tr>
<tr><td>posix_getgid()</td><td>It accept group ID returned by filegroup() function as an argument and returns an associative array on a group identified by group ID with following refernces
<table>
<tr><td>Key</td><td>Description</td></tr>
<tr><td>Name</td><td>The name of group</td></tr>
<tr><td>Gid</td><td>The ID number of group</td></tr>
<tr><td>members</td><td>The number of members belonging to the group</td></tr>
</table></td></tr>
<tr><td>filetype()</td><td>It takes filename as an argument and returns the type of specified file . the type of possible values are fifo, char, dir, block, link, file and unknown</td></tr>
<tr><td>basename()</td><td>It takes file name as an argument and separate the filename from its directory path.</td></tr>
<tr><td>copy()</td><td>It takes two string arguments referring to the source and destination file respectively.</td></tr>
<tr><td>rename()</td><td>It takes two argument as old name and new name and renames the file with new name.</td></tr>
<tr><td>unlink()</td><td>It takes a single argument referring to the name of file we want to delete.</td></tr>
<tr><td>is_file()</td><td>It returns true if the given file name refers to a regular file.</td></tr>
<tr><td>fstat()</td><td>The fstat() function returns information about an open file.</td></tr>
</table>

	This function returns an array with the following elements: [0] or [dev] - Device number [1] or [ino] - Inode number [2] or [mode] - Inode protection mode [3] or [nlink] - Number of links [4] or [uid] - User ID of owner [5] or [gid] - Group ID of owner [6] or [rdev] - Inode device type [7] or [size] - Size in bytes [8] or [atime] - Last access (as Unix timestamp) [9] or [mtime] - Last modified (as Unix timestamp) [10] or [ctime] - Last inode change (as Unix timestamp) [11] or [blksize] - Blocksize of filesystem IO (if supported) [12] or [blocks] - Number of blocks allocated

Examples

Use of some above mentioned functions is illustrated in the followingexamples:

Example : 1) To read file from server use fread() function. A file pointer can be created to the file and read the content by specifying the size of data to be collected.

```
 <?php
$myfile = fopen("somefile.txt", "r") or die("Unable to open file!");
echo fread($myfile,filesize("somefile.txt"));
fclose($myfile);
?>
```

Example : 2) a file can be written by using fwrite() function in php. for this open file in write mode. file can be written only if it has write permission. if the file does not exist then one new file will be created. the file the permissions can be changed.

```
<?php
$filecontent="some text in file";          // store some text to enter inside the file
$file_name="test_file.txt";                // file name
$fp = fopen ($filename, "w");              // open the file in write mode, if it does not
exist then it will be created.
fwrite ($fp,$filecontent);                 // entering data to the file
fclose ($fp);                              // closing the file pointer
chmod($filename,0777);                     // changing the file permission.
?>
```

Example : 3) A small code for returning a **file-size.**

```
<?php
function dispfilesize($filesize){
   if(is_numeric($filesize))
{        $decr = 1024; $step = 0;
         $prefix = array('Byte','KB','MB','GB','TB','PB');
         while(($filesize / $decr) > 0.9)
{        $filesize = $filesize / $decr;
         $step++;
```

```
        }
    return round($filesize,2).' '.$prefix[$step];
    } else
        {    return 'NaN';
        }
}
?>
```

Example : 4) Print file's extension of the given file

```
<?php
         $file = $_FILES['userfile'];
         $allowedExt = array("txt", "rtf", "doc");
         function isAllowedExtension($fileName)
{         global $allowedExt;
                 return in_array(end(explode(".", $fileName)), $allowedExt);
 }
 if($file['error'] == UPLOAD_ERR_OK) {
 if(isAllowedExtension($file['name'])) {
 } else {
                 echo "Invalid file type";
  }} else die("Cannot upload");
?>
```

Working with Directories

A **directory** is special type of file that holds the names of other files and directories and pointer to their storage area on media. A **directory handle** is nothing more than an integer value pointing to a directory ,which can be obtained bf specifying the directory in call to the opendir() function.

Function Name	Purpose
opendir()	It takes directory name with detail path as an argument and returns directory handle on success , both otherwise false.
closedir()	It takes directory handle as an argument and close directory
readdir()	It takes directory handle as an argument and returns the next entry listed in the open directory.
Other directory functions	
rewinddir()	It accepts a directory handle as an argument and reset the corresponding directory position indicator to the beginning of the directory
chdir()	This function changes current directory to given directory
rmdir()	It remove specified directory
mkdir()	It creates directory as specified in its first argument
dirname()	It returns directory part of given file name
is_dir()	It returns true if the given file name refers to a directory.

Examples

Use of some of the above mentioned functions related to the directory is illustrated in the following examples:

Example : 1) Program for directory traversal and printing files and

```
$handle=opendir(".");                    // open the current directory by opendir
while (($file = readdir($handle))!==false) {
echo "$file <br>";     }
closedir($handle);
```

Example : 2) Find filenames with .php extension.

```
$path="../dir-name/";                    // path of the directory
$handle=opendir($path);
while (($file_name = readdir($handle)) != false) {                 // read the file
if(stristr($file_name, ".php"))
                echo $file_name;
}
```

<u>Set A</u>

Q: 1) Write a program to read two file names from user and append contents of first file into second file.

Q: 2) Write program to read directory name from user and display content of the directory.

<u>Set B</u>

Q: 1) Write a program to read a flat file “student.dat”, calculate the percentage and display the data from file in tabular format.(Student.dat file contains rollno, name, Syspro, TCS, CN, PHP, JAVA, BA)

Q: 2) Write a program to read directory name and extension. Display the files with specified extension from that directory.

<u>Set C</u>

Q: 1) Write a menu driven program to perform various file operations.

a) Display size of file
b) Display Last Access, changed, modified time of file
c) Display details about owner and user of File
d) Display type of file
e) Delete a file
f) Copy a file
g) Traverse a directory in hierarchy
h) Remove a directory

Q: 2) Write a program to read directory name from user and display content of the directory recursively.

Signature of the instructor :____________________ Date :_____________

Assignment Evaluation

0:Not Done □ 2:Late Complete □ 4:Complete □

1:Incomplete □ 3:Needs Improvement □ 5:Well Done □

ASSIGNMENT NO. 4 : OBJECT ORIENTED PROGRAMMING

Class :A class is a unit of code composed of variables and functions which describes the characteristics and behavior of all the members of the set.

Function	Description	Example

class classname [extends baseclass]	Creates a class	Class student
		{
		[var $propery [= value];…]
		[function functionname (arguments)
		{
		//code
		}
		….
		]
		}

$instance = new classname();	Create an object	<?php $instance1 = new myclass (); //This can also be done with a variable: $newname= ‘hello’; $instance2 = new $newname(); ?>

class classname
{
function methodname()
{
Statements;
}
}
	Add a Method	<?php
		class myclass
		{ function mymethod()
		{ print “ hello, myclass}}
		?>
		To invoke the method on the object $instance1, we need to invoke the operator “->" to access the newly created function mymethod
		
		<?php
		$instance1=new myclass();
		$instance1->mymethod();
		?>

public $publicMemeber = “Public member”;	Adding Property
	
	Public
		Public :
		<?php
		class maths
		{
		public $num;
		public function multi()
		{
		return $this->num*2;
		}
		}

		$math=new maths; $math->num=2; echo $math->multi(); ?> Output : 4

protected $protectedmember = “Protected Member”;
Private $privatemember= “Private Member
	Protected
	Private
		Protected:
		
		<?php
		class maths
		{
		protected $num;
		public function setnum($num)
		{
		    $this->num=$num;
		}
		
		public function multi()
		{
		    return $this->num*2;}}
		
		class add extends maths
		{
		    public function addtwo()
		    {
		     $new_num=$this->num + 2;
		      return ($new_num);
		    }
		}
		$math=new add;
		$math->setnum(14);
		echo $math->addtwo();
		?>
		
		Output : 16

class extendedClass extends classname	Inheritance
	It is the ability of PHP to extend classes that inherit the characteristics of the parent class.
	
	It is not possible to extend multiple classes ; a class can only inherit from one base class.
		<?php
		class myclass
		{
		//property declaration
		public $var=’a default value’;
		
		//method declaration
		public function desplayVar()
		{
		    echo $this->var;
		}
		}

		class extendedClass extends myclass
		{
		//redefine the parent method
		function displayVar()
		{
		echo “Extending Class”;
		parent::displayVar();
		}
		}
		
		$extend =new extendedClass();
		$extend->displayVar();
		?>
		
		Output :
		Extending class
		a default value

Overriding	When we give a function in the child class the same name as a function in the parent class, this concept is called function overriding.
	
	Any method or class that is declared as final can not be overridden or inherited by another class.
		<?php
		class Hello
		{function getMessage()
		{ return ‘Hello World !’;}
		}
		class Goodbye extends Hello
		{function getMessage(){
		return ‘Goodbye World!’;}}
		$hello=&new Hello();
		Echo $hello->getMessage().’<br/>’;
		$goodbye = &new Goodbye();
		Echo $goodbye->getMessage().
		’<br/>’;?>
		Output:
		Hello World!
		Goodbye World!

void _construct ([mixed $args [, $....]])	Constructor is a function which is called right after a new object is created.	<?php
		class Student
		{
		var $name;
		var $address;
		var $phone;
		
		//This is constructor
		function student()
		{
		this->name=”abc”;
		this->address=”pqr”;
		this->phone=1111;
		}
		
		function printstudentinfo()

		{
		echo this->name . ”\n”;
		echo this->address . ”\n”;
		echo this->phone . “\n”;
		}
		}
		$stud =new student();
		$stud->printstudentinfo();
		$stud=NULL;
		?>

void _destruct (void)	Destructor is a function which is called right after you release an object.	<?php
		class Student
		{
		var $name;
		var $address;
		var $phone;
		
		//This is constructor
		function _construct()
		{
		this->name=”abc”;
		this->address=”pqr”;
		this->phone=1111;
		}
		
		function _destruct()
		{
		echo “Student Object Released”;}
		
		function printstudentinfo()
		{
		Echo this->name . ”\n”;
		echo this->address . ”\n”;
		echo this->phone . “\n”;
		}
		}
		$stud =new student();
		$stud->printstudentinfo();
		$stud=NULL;
		?>

class_exist()	Introspection We can use this function to determine whether a class exists.	$class = class_exists(classname);
get_declared_classes()	This function returns array of defined classes and checks if the class name is in returned array.	$classes = get_declared_classes();
get_class_methods()	We can use this function to get the methods and properties of class	$methods = get_class_methods(classname);
get_class_vars()	This function returns only properties that have default values.	$properties=get_class_vars(classname);
get_parent_class()	This function is used to find the class's parent class.	$superclass = get_parent_class (classname);
is_object()	Is_object function is used to make sure that it is object.	$obj= is_obj(var);
get_class()	get_class() function is used to get the class to which an object belongs and to get class name	$classname= get_class(object);
method_exists()	This function is used to check if method on an object exists .	$method_exists=method_exists(object ,method);
get_object_vars()	This function returns an array of properties set in an object	$array=get_object_vars(object);
serialize()	Serialization Serializing an object means converting it to a byte stream representation that can be stored in a file. returns a string containing a byte-stream representation of the value that can be stored anywhere	$encode=serialize(something)
unserialize()	Takes a single serialized variable and converts it back to PHP value.	$something = unserialize (encode);

Interfaces	An interface is declared similar to a class but only include function prototypes (without implementation) and constants. When a class uses an interface the class must define all the methods / function of the interface otherwise the PHP engine will give you an error.
	
	The interface’s function /methods cannot have the details filled in. that is left to the class that uses the interface.
		Example of an interface
		class duck
		{ function quack()
		{
		echo “quack,quack,qk, qk…”;
		}
		}
		Interface birds
		{
		function breath();
		function eat();
		}
		Class duck implements birds
		{ function quack()
		{
		echo “quack,quack,qk, qk…”;
		}
		}
		function breath()
		{
		echo “duck is breathing”;
		}
		function eat()
		{
		echo “ duck is eating”;
		}

Encapsulation	Encapsulation is an ability to hide details of implementation.	<?php
		class A
		{
		function check()
		{
		if(isset ($this))
		{
		echo “$this is defined (“;
		echo get_class($this);
		echo “)\n”;
		}
		else
		{
		echo “this is not defined”;
		}
		}
		}
		class B
		{
		function bcheck()
		{
		A::check();
		}

		} $a=new A(); $a->check(); A::check(); $b=new B(); $b->bcheck(); B::bcheck(); ?> Output: $this is defined(a) $this is not defined $this is defined(b) $this is not defined

Set A

Q: 1) Define an interface which has methods area( ), volume( ). Define constant PI. Create a class cylinder which implements this interface and calculate area and volume. (Hint: Use define( ))

Q: 2) Write class declarations and member function definitions for an employee(code, name, designation). Derive emp_account(account_no, joining_date) from employee and emp_sal(basic_pay, earnings, deduction) from emp_account.

Write a menu driven program

a) To build a master table
b) To sort all entries
c) To search an entry
d) Display salary
e)

Set B

Q:1) Create class rectangle and derive a class square from class Rectangle. Create another class circle. Create an interface with only one method called area(). Implement this interface in all the classes. Include appropriate data members and constructors in all classes. Write a program to accept details of a square, circle and rectangle and display the area.

Q:2) Create a class account(accno,cust_name). Derive two classes from account as saving_acc(balance, min_amount) and current_acc(balance, min_amount).

a) Display a menu
b) Saving Account
c) Current Account

For each of this display a menu with the following options.

1. Create account
2. Deposit
3. Withdrawal

Set C

Q:1) Define an interface for stack operation. Implement this interface in a class.

Q:2) Write necessary class and member function definitions for a cricket player object.The program should accept details from user (max :10) (player_code, name, runs, innings_played, no_of_times_out).
The program should contain following menu.
Enter details of players.
Display average runs of a single player.
Average runs of all players.
Display the list of players in sorted order as per runs(use function overloading)

Signature of the instructor :____________________ Date :_____________

Assignment Evaluation

0:Not Done	□	2:Late Complete	□	4:Complete	□
1:Incomplete	□	3:Needs Improvement	□	5:Well Done	□

ASSIGNMENT NO. 5 : PHP-DATABASE(POSTGRESQL)

PostgreSQL supports a wide variety of built-in data types and it also provides an option to the users to add new data types to PostgreSQL, using the CREATE TYPE command. Table lists the data types officially supported by PostgreSQL. Most data types supported by PostgreSQL are directly derived from SQL standards. The following table contains PostgreSQL supported data types for your ready reference

Category	Data type	Description
Boolean	boolean, bool	A single true or false value.
Binary types	bit(n)	An n-length bit string (exactly n) binary bits)
	bit varying(n), varbit(n)	A variable n-length bit string (upto n) binary nbits)
Character Types	character(n)	A fixed n-length character string
	char(n)	A fixed n-length character string
	character varying(n)	
	varchar (n)	
	text	A variable length character string of unlimited length
Numeric types	smallint, int2	A signed 2-byte integer
	integer, int, int4	A signed, fixed precision 4-byte number
	bigint, int8	A signed 8-byte integer, up to 18 digits in length
	real, float4	A 4-byte floating point number
	float8, float	An 8-byte floating point number
	numeric(p,s)	An exact numeric type with arbituary precision p, and scale s.
Currency	money	A fixed precision, U.S style currency
	serial	An auto-incrementing 4-byte integer
Date and time types	date	The calendar date(day, month and year)
	time	The time of day
	time with time zone	the time of day, including time zone information
	timestamp(includes time zone)	

	Interval)	An arbitrarily specified length tttottimtime

Functions used for postgreSQL database manipulation

Function name	**Purpose**	**Example**
resource **pg_connect** (string $connection_string [, int $connect_type]);	Open a PostgreSQL connection	$conn = pg_connect("host", "port", "options", "tty", "dbname");
resource **pg_pconnect** (string $connection_string [, int $connect_type]);	Open a persistent PostgreSQL connection	$conn_string = "host=sheep port=5432 dbname=test user=lamb password=bar";
resource **pg_prepare** ([resource $connection], string $stmtname , string $query)	Submits a request to create a prepared statement with the given parameters, and waits for completion.	$result = pg_prepare($dbconn, "my_query", 'SELECT * FROM shops WHERE name = $1');
resource **pg_execute** ([resource $connection], string $stmtname , array $params)	Sends a request to execute a prepared statement with given parameters, and waits for the result.	$result = pg_execute($dbconn, "my_query", array("Joe's Widgets"));

resource **pg_query** ([resource $connection], string $query)	Execute a query	$result = pg_query($conn, "SELECT author, email FROM authors");
		if (!$result) {
		  echo "An error occurred.\n";
		  exit;
		}

array **pg_fetch_assoc** (resource $result [, int $row])	Fetch a row as an associative array	while ($row = pg_fetch_assoc($result)) echo $row['id'];
bool **pg_close** ([resource $connection])	Closes a PostgreSQL connection	pg_close($dbconn);

Example to create php Postgrsql Connectivity and display records

```
<?php
$conn = pg_connect("dbname=publisher");
if (!$conn) {
  echo "An error occurred.\n";
  exit;
}

$result = pg_query($conn, "SELECT id, author, email FROM authors");
if (!$result) {
  echo "An error occurred.\n";
  exit;
}

while ($row = pg_fetch_assoc($result)) {
  echo $row['id'];
  echo $row['author'];
  echo $row['email'];
}
?>
```

Set A

Q: 1) Consider the following entities and their relationships

Emp (emp_no,emp_name,address,phone,salary)
Dept (dept_no,dept_name,location)

Emp-Dept are related with one-many relationship Create a RDB in 3NF for the above and solve following

Using above database write a PHP script which will print a salary statement in the format given below, for a given department. (Accept department name from the user).

Deparment Name : _________________

Maximum Salary	Minimum Salary	Sum Salary

Q: 2) Consider the following entities and their relationships
Doctor (doc_no, doc_name, address, city, area)
Hospital (hosp_no, hosp_name, hosp_city)
Doctor and Hospital are related with many-many relationship. Create a RDB in 3 NF for the above and solve following

Using above database, write a PHP script which accepts hospital name and print information about doctors visiting / working in that hospital in tabular format.

Set B

Q: 1) Considerer the following entities and their relationships
project(pno integer, p_name char(30), ptype char(20),duration integer)
employee (eno integer, e_name char (20), qualification char (15), joindate date)
The relationship between project - employee: M-M, with descriptive attributes as start_date (date), no_of_hours_worked (integer).

Using above database write a script in PHP to accept a project name from user and display information of employees working on the project.

Q: 2) Consider the following entities and their relationships
student (sno integer, s_name char(30), s_class char(10), s_addr char(50))
teacher (tno integer, t_name char (20), qualification char (15),experience integer)
The relationship between student-teacher: m-m with descriptive attribute subject.

Using above database write a script in PHP to accept a teacher name from user and display the names of students along with subjects to whom teacher is teaching

---

Set C

Q: 1) Consider the following entities and their relationships

Movie (movie_no, movie_name, release_year)
Actor (actor_no, name)

Relationship between movie and actor is many – many with attribute rate in Rs. Create a RDB in 3 NF for the above and solve following

Using above database, write PHP scripts for the following:(Hint: Create HTML form having three radio buttons)

a) Accept actor name and display the names of the movies in which he has acted.
b) Insert new movie information.
c) Update the release year of a movie. (Accept the movie name from user)

Q: 2) Considerer the following entities and their relationships

Student (Stud_id,name,class)
Competition (c_no,c_name,type)

Relationship between student and competition is many-many with attribute rank and year. Create a RDB in 3NF for the above and solve the following.

Using above database write a script in PHP to accept a competition name from user and display information of student who has secured 1st rank in that competition.

Signature of the instructor :____________________ Date :_____________

Assignment Evaluation

0:Not Done	□	2:Late Complete	□	4:Complete	□
1:Incomplete	□	3:Needs Improvement	□	5:Well Done	□

Project Work Guidelines

PREPARED BY:

Ms.Madhuri Deshpande(S P College)
Ms. Kalpana Joshi(Fergusson College)
Ms. Shilpa Khadilkar(Kaveri College)

The project will be done in following environment:
Operating System : ***Linux***
Programming Language used : ***Java / PHP***
Database : ***PostgreSQL***
Marks will be given as follows

Sr. No.	**Task**	**Marks**	**Sign**
1.	Problem Definition	/ 5	
2.	Proposed system	/ 5	
3.	Feasibility study	/ 5	
4.	Gathering Data Requirements and Functional Requirement	/ 5	
5.	Designing the normalized Database	/ 5	
6.	UML Diagrams	/ 5	
7.	I/O screens	/ 5	
8.	Test Case Design	/ 5	
9.	Coding	/10	
	Total	**/50**	
	Convert above 50 to : Total	**/20**	
10.	Final Demo (to be considered in Final Practical Examination)	**/10**	

The 50 marks will be converted to 20 marks as INTERNAL Marks.

The final demo marks 10 to be given separately. These marks will be considered in Final Examination in the 30 marks.

1. Internal Examiner will keep both the mark lists (Internal 20 marks, 10 Marks Final Demo) ready before examination.
2. Include table in the lab book of PHP.
3. The copy of progress report will be put in the final project report for reference.
4. The database design will be applicable to database related projects.
5. The Game based project will have the complete life cycle of that game along with Win and Loss conditions.
6. We can draw only applicable and relevant UML diagrams and not all.

Format of Project Progress Report

Roll No & Name of the student	
Title of the Project	
Project Guide Name	

Sr. No.	Date	Task done	Sign (with date)

Head,

Dept. of Computer Science

COMPUTER NETWORK ASSIGNMENTS

Assignments based on CS-333 (Computer Networks -I) and CS-343 (Computer Networks -II)

PREPARED BY:

PROF. MS. POONAM PONDE (NOWROSJEE WADIA COLLEGE)

PROF. JEEVAN LIMAYE (FERGUSSON COLLEGE)

- **OBJECTIVES**

The objectives of these assignments are :

a. To cover basic concepts of networking

b. To understand how networking protocols work

c. To understand basic Linux installation and setting up of the operating environment

d. To study LAN setup and understand basic LAN principles

e. To study tools for network analysis

Assignment Completion Sheet

Sr. No	Assignment Name	Marks
1	Linux Installation	
2	Networking commands in Linux	
3	Study of LAN environment	
4	Use of Wireshark tool	
	Total out of 20	
	Total out of 10	

Signature of Incharge :

Assignment 1 : Linux Installation and operating environment

Instructors should demonstrate :

1. Linux installation
2. Creating users
3. Creating user groups
4. Setting permissions for home directory of users
5. Important files and directories in linux and their use
6. Configuring Apache server and Apache Tomcat
7. Configuring database using postgresql

Self study questions for students :

1. List the stages of Linux boot process
2. What is runlevel? What are the predefined runlevels?
3. Find out the runlevel of your computer
4. Find out the kernel version of your machine
5. What is NIS and NFS ?
6. What is the use of RPM ? List various options of rpm command with syntax
7. State the purpose of the following files and directories:
    a. /home
    b. /boot
    c. /dev
    d. /usr
    e. /mnt
    f. /media
    g. /etc
    h. /bin
    i. /usr/bin
    j. /etc/fstab
    k. .bashrc

Signature of the instructor □ Date □

Assignment Evaluation

0: Not done □ 2: Late Complete □ 4: Complete □

1: Incomplete □ 3: Needs improvement □ 5: Well Done □

Assignment 2: Networking commands in Linux

Execute the following commands and write their output

1. ping :

This command is used to test connectivity between two nodes. Ping use ICMP (Internet Control Message Protocol) to communicate to other devices. You can ping host name or ip address using below command.

example: ping 201.54.100.1 or ping www.google.com

$ping <server-ip-address>
Output:
$ping localhost
Output:
$ping <other-ip-in-network>
Output:

2. hostname

Gives the host name of the computer they are logged into. To set the hostname permanently use /etc/sysconfig/network file.

$hostname
Output :

3. traceroute

traceroute is a network troubleshooting utility which shows number of hops taken to reach destination also determine packets traveling path.

$traceroute ip-address
Output :

4. netstat

Netstat (**Network Statistic**) command displays interfaces, connection information, routing table information etc.

$netstat
Output :

Execute it with the following options and write the output:

netstat –t

netstat –s –t

netstat -i

5. ifconfig

ifconfig is used for displaying network interface information.

$/sbin/ifconfig
Output :

6. who

Displays information of all users who are logged in

$who
Output :

7. whoami

The whoami command writes the user name (i.e., login name) of the owner of the current login session to standard output.

$whoami
Output :

8. nmap

Network mapper tool to discover hosts and services on a computer network.

$ nmap <ip-address>
Output :

$ nmap <server-ip-address>
Output :

9. tcpdump

Tcpdump prints out a description of the contents of packets on a network interface that match the boolean expression; the description is preceded by a time stamp, printed, by default, as hours, minutes, seconds, and fractions of a second since midnight.

Sample output for ARP protocol:

```
arp who-has 128.3.254.6 tell 128.3.254.68

arp reply 128.3.254.6 is-at 02:07:01:00:01:c4
```

$ tcpdump
Output :

Signature of the instructor □ Date □

Assignment Evaluation

0: Not done □ 2: Late Complete □ 4: Complete □

1: Incomplete □ 3: Needs improvement □ 5: Well Done □

Assignment 3 : Study of LAN environment

Find out information about the network in your lab and fill in details below:

1. Total Number of computers in your lab:
2. Find details of any 5 computers :

	MAC address	IP address	LAN speed	Default mask	hostname
1					
2					
3					
4					
5					

3. Are the IP addresses assigned to the machines statically or dynamically?
4. Does the network have a DHCP server?
5. If yes, what is the address of the server ?
6. How many servers are configured? :

   Details of servers :

	IP address	MAC address	Purpose
1			
2			
3			

7. Cables
   a. Type :
   b. Is it coaxial / twisted pair or fiber optic cable ?
   c. Cable bandwidth
   d. Maximum cable length limit
   e. Connector used
8. Switches:

No	Company	MAC address	No. of	Managed /	IP's of

	Name		ports	Unmanaged	Machines connected to the switch
1					
2					
3					
4					
5					

9. Routers:

No	Company Name	No. / Types of ports	Port speed	IP address	
1					
2					
3					

10. Is there wi-fi capability in the LAN?
    If yes,
    i. What is the Wi-fi access point address?
    ii. How many devices / IP addresses does it support?
    iii. What is the bandwidth?
    If no,
    iv. What additional devices are needed?
    v. Where will you connect them?
    vi. What will be its IP address?

11. Is there internet access in the lab?
    If not, what changes to the hardware / software must be made ?
    If yes, what is the IP address of the router / gateway ?

12. Draw the Network Topology (show how machines and servers are connected using connectivity devices)

13. If 20 more machines have to be added to the network, what changes must be made to the network?

14. If the network is to be divided into four subnetworks having 50 machines each, give a plan to do so. What additional devices will be needed ? Give the IP address of each subnetwork and the address ranges for hosts in each subnetwork.

Signature of the instructor □ Date □

Assignment Evaluation

0: Not done □ 2: Late Complete □ 4: Complete □

1: Incomplete □ 3: Needs improvement □ 5: Well Done □

Assignment 4 : Use of Wireshark tool

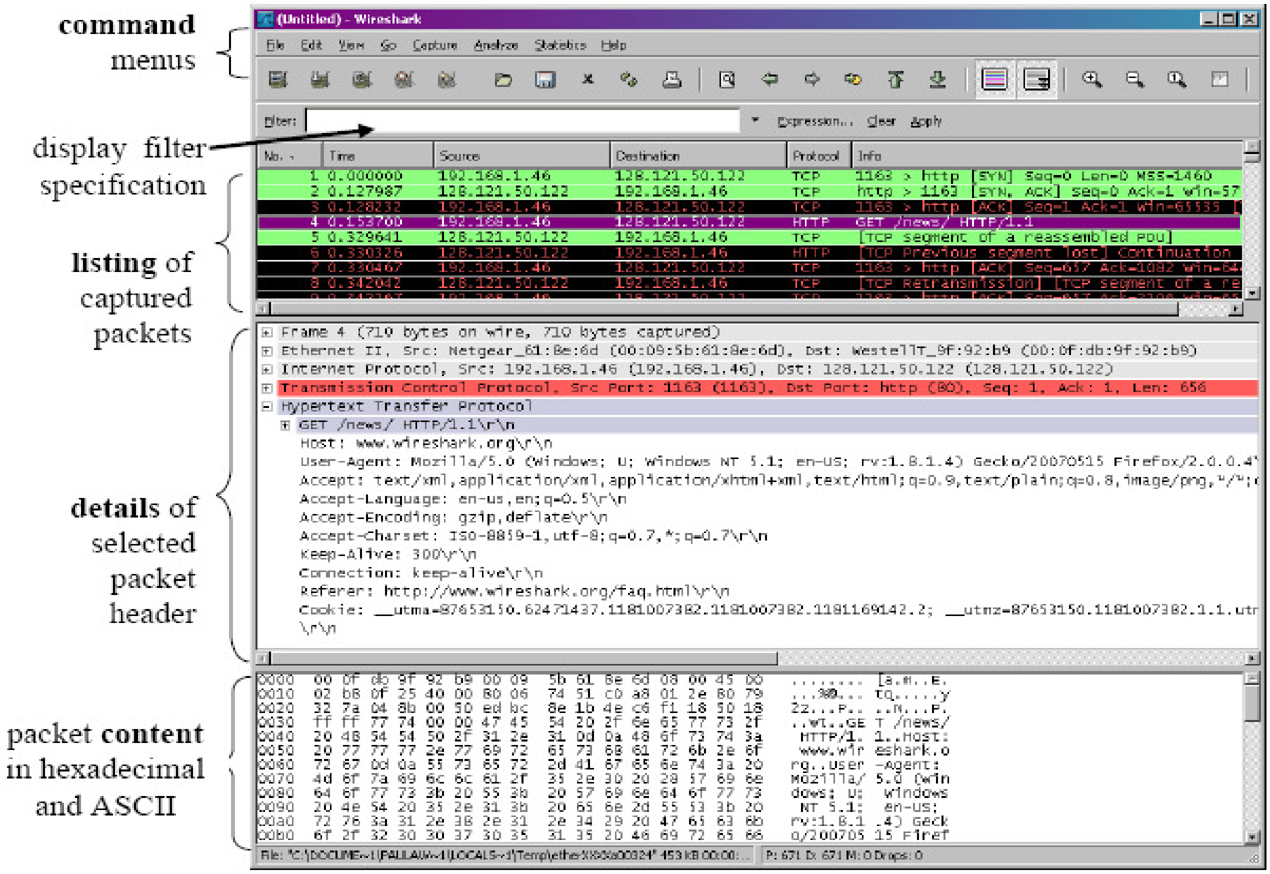
Demonstrate the use of Wireshark tool for network analysis

Wireshark is a free and open source packet analyzer. It is also a protocol analyzer tool which captures network traffic and analyzes it. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named Ethereal, the project was renamed Wireshark in May 2006 .

Purpose:

- network administrators use it to **troubleshoot network problems**
- network security engineers use it to **examine security problems**
- developers use it to **debug protocol implementations**
- people use it to **learn network protocol** internals

The wireshark GUI is as shown :


command menus
display filter specification
listing of captured packets
details of selected packet header
packet content in hexadecimal and ASCII
(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No. -  Time        Source           Destination      Protocol  Info
     1 0.000000    192.168.1.46     128.121.50.122   TCP       1163 > http [SYN] Seq=0 Len=0 MSS=1460
     2 0.127987    128.121.50.122   192.168.1.46     TCP       http > 1163 [SYN, ACK] Seq=0 Ack=1 Win=57
     3 0.128232    192.168.1.46     128.121.50.122   TCP       1163 > http [ACK] Seq=1 Ack=1 Win=65535 [
     4 0.153700    192.168.1.46     128.121.50.122   HTTP      GET /news/ HTTP/1.1
     5 0.329641    128.121.50.122   192.168.1.46     TCP       [TCP segment of a reassembled PDU]
     6 0.330326    128.121.50.122   192.168.1.46     HTTP      [TCP Previous segment lost] Continuation
     7 0.330467    192.168.1.46     128.121.50.122   TCP       1163 > http [ACK] Seq=657 Ack=1082 Win=64
     8 0.342042    128.121.50.122   192.168.1.46     TCP       [TCP Retransmission] [TCP segment of a re
```

```
⊞ Frame 4 (710 bytes on wire, 710 bytes captured)
⊞ Ethernet II, Src: Netgear_61:8e:6d (00:09:5b:61:8e:6d), Dst: WestellT_9f:92:b9 (00:0f:db:9f:92:b9)
⊞ Internet Protocol, Src: 192.168.1.46 (192.168.1.46), Dst: 128.121.50.122 (128.121.50.122)
⊞ Transmission Control Protocol, Src Port: 1163 (1163), Dst Port: http (80), Seq: 1, Ack: 1, Len: 656
⊟ Hypertext Transfer Protocol
  ⊞ GET /news/ HTTP/1.1\r\n
    Host: www.wireshark.org\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4
    Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Referer: http://www.wireshark.org/faq.html\r\n
    Cookie: __utma=87653150.62471437.1181007382.1181007382.1181169142.2; __utmz=87653150.1181007382.1.1.utr
    \r\n
```

```
0000  00 0f db 9f 92 b9 00 09  5b 61 8e 6d 08 00 45 00   ........ [a.m..E.
0010  02 b8 0f 25 40 00 80 06  74 51 c0 a8 01 2e 80 79   ...%@... tQ.....y
0020  32 7a 04 8b 00 50 ed bc  8e 1b 4e c6 f1 18 50 18   2z...P.. ..N...P.
0030  ff ff 77 74 00 00 47 45  54 20 2f 6e 65 77 73 2f   ..wt..GE T /news/
0040  20 48 54 54 50 2f 31 2e  31 0d 0a 48 6f 73 74 3a    HTTP/1. 1..Host:
0050  20 77 77 77 2e 77 69 72  65 73 68 61 72 6b 2e 6f    www.wir eshark.o
0060  72 67 0d 0a 55 73 65 72  2d 41 67 65 6e 74 3a 20   rg..User -Agent: 
0070  4d 6f 7a 69 6c 6c 61 2f  35 2e 30 20 28 57 69 6e   Mozilla/ 5.0 (Win
0080  64 6f 77 73 3b 20 55 3b  20 57 69 6e 64 6f 77 73   dows; U;  Windows
0090  20 4e 54 20 35 2e 31 3b  20 65 6e 2d 55 53 3b 20    NT 5.1;  en-US; 
00a0  72 76 3a 31 2e 38 2e 31  2e 34 29 20 47 65 63 6b   rv:1.8.1 .4) Geck
00b0  6f 2f 32 30 30 37 30 35  31 35 20 46 69 72 65 66   o/200705 15 Firef
```

File: "C:\DOCUME~1\PALLAW~1\LOCALS~1\Temp\etherXXXXa00324" 453 KB 00:00:... P: 671 D: 671 M: 0 Drops: 0

1. Capture and view network traffic
2. Look at the packet headers of various protocols
3. View the detailed contents of the following packets in hexadecimal.
    - i. Ethernet
    - ii. IP
    - iii. TCP
    - iv. ARP
4. Write the contents of Ethernet frame header and list down the values of all fields in the header.

Header values in hexadecimal

- i. What is the Source MAC address
- ii. What is the destination MAC address
- iii. Is the destination MAC address of the server?
- iv. What is the value of CRC field?
- v. What is the destination MAC address of the Ethernet frame containing an ARP request?

5. View protocol hierarchy statistics

6. Follow TCP stream

Signature of the instructor □ Date □

Assignment Evaluation

0: Not done □	2: Late Complete □	4: Complete □
1: Incomplete □	3: Needs improvement □	5: Well Done □