

**Class Note 6**  
**PHP Oracle Development**  
**(Updated 7/2/2015)**

**Oracle Connection**

The “class note” is the typical material I would prepare for my face-to-face class. I am sharing the notes with everyone assuming you are in the class.

For the best result of your study, please make sure you do the following:

1. Read the chapter(s) in your textbook, corresponding to the week.
2. Read this class note. Please understand that I will only highlight the important points from your chapters in the textbook. I do not intend to repeat it.

Now, here comes the most exciting part of this course: how to use PHP to access your Oracle database so that you may present or manipulate the data on or through your web page? Before you are able to do this, the first step is, of course, you have to have an Oracle database up and running.

- I. Version(s) of Oracle Database
  - a. Oracle Enterprise Edition (Oracle 9i, 10g, 11g or 12c): If you have taken my TEC 5333 Advanced Database Technology or TEC 5363 Database Security and Reliability, chances are you have already had Oracle Enterprise Edition (9i, 10g, 11g or 12c) installed in your computer. You may just use the Oracle database as it is installed. In other words, if you have Oracle installed in your computer already, you may just use it. No more sweat is necessary. If you are patient enough, I will show you the trick to connect to the database in the later part of this Class Notes.
  - b. If you have not installed Oracle database on your computer, you may select any of Oracle database versions (Oracle 9i, 10g, 11g or 12c). If you have an old computer like my old laptop (XP, 256MB RAM), Oracle 10g or 11g Express Edition will be ideal since it takes the least amount of resources to run it. It does not slow down your computer. In other words, any Oracle database will suffice for this class. If you have never done anything with Oracle before, I recommend Oracle 10g or 11g Express Edition (as in your textbook).
  - c. If you have never done the installation of Oracle database, the first question you would have is how do I can get the Oracle database. The answer is: you need to download it from Oracle’s web site. EIU has a corporate license agreement with Oracle. Thus, you may be able to download a whole range of Oracle products. The one you need will be Oracle database (in any version of your choice) that is designed for your computer platform

(Windows, Linux, Unix, or Mac). You may have to register with Oracle. That is just one hurdle you have to jump before you can download. There is no charge.

If you are lazy and do not even want to type "Oracle 10g Express Edition" on Google search, you may use the following link to find the Oracle Express Edition:

<http://www.oracle.com/technology/software/products/database/xe/index.html>

## II. Installation of Oracle Database

The detailed installation guide for Oracle 9i, 10g or 11g Enterprise Edition can be available in the following course web sites for my previously mentioned classes.

<http://www.eiu.edu/~pingliu/tec5323/Schedule/tec5323schedule.htm>

or

<http://www.eiu.edu/~pingliu/tec5363/Schedule/tec5363Schedule.htm>

The following is my notes for my Oracle Express Edition installation, for your reference.

Notes for Oracle XE installation (on my old laptop):

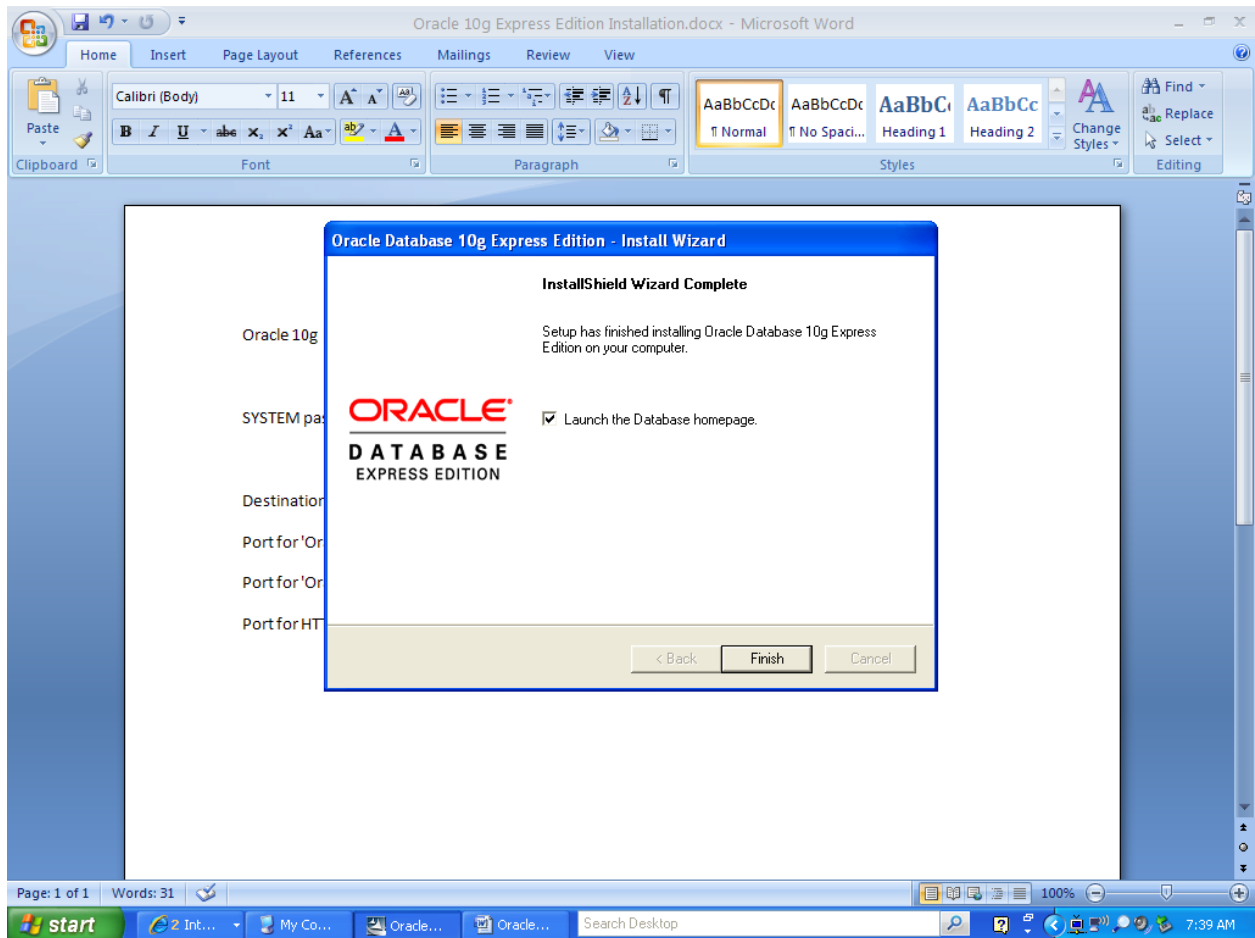
Destination Folder: C:\oraclexe\

Port for 'Oracle Database Listener': 1521

Port for 'Oracle Services for Microsoft Transaction Server': 2030

Port for HTTP Listener: 8080

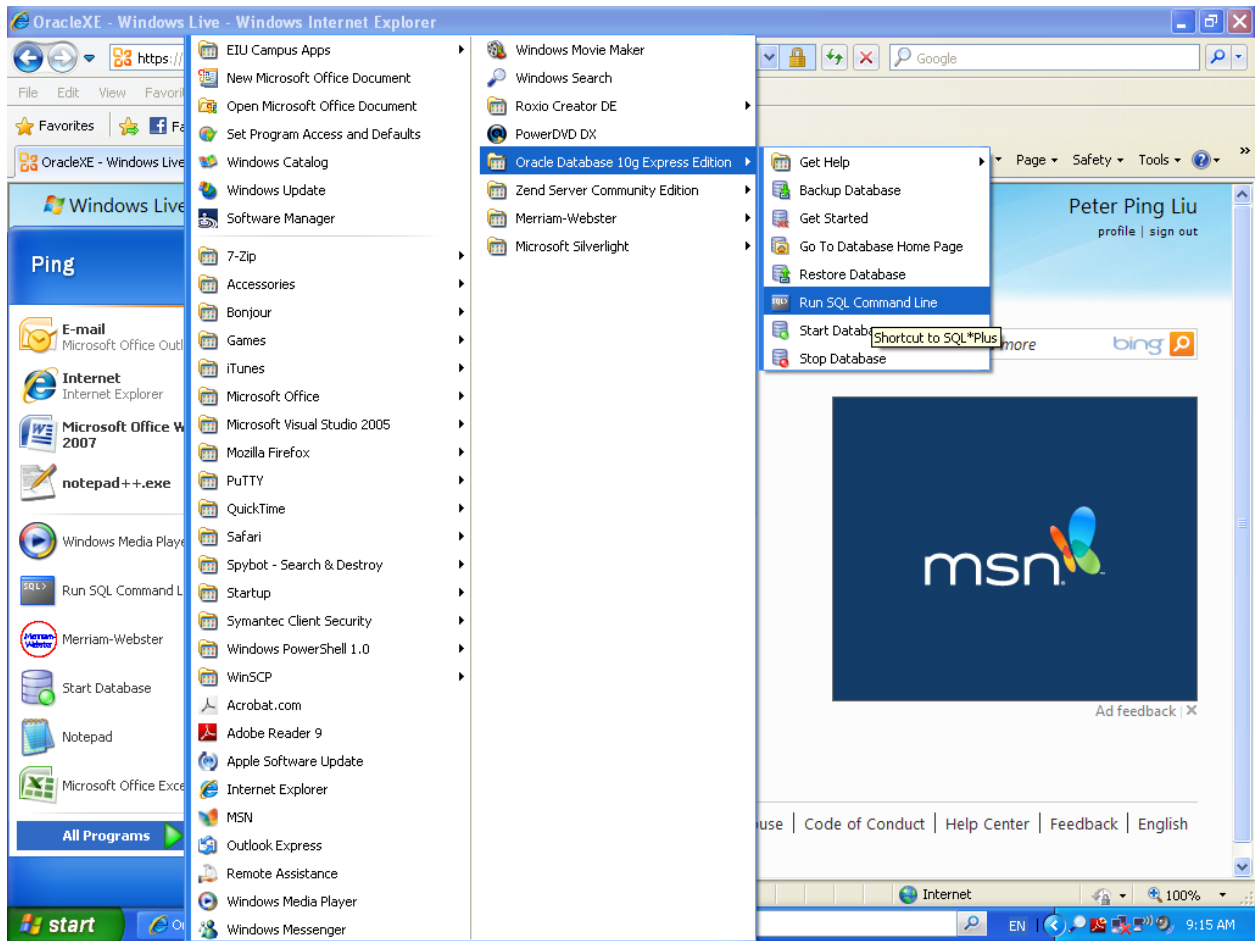
Please make sure to take note on you SYSTEM password.



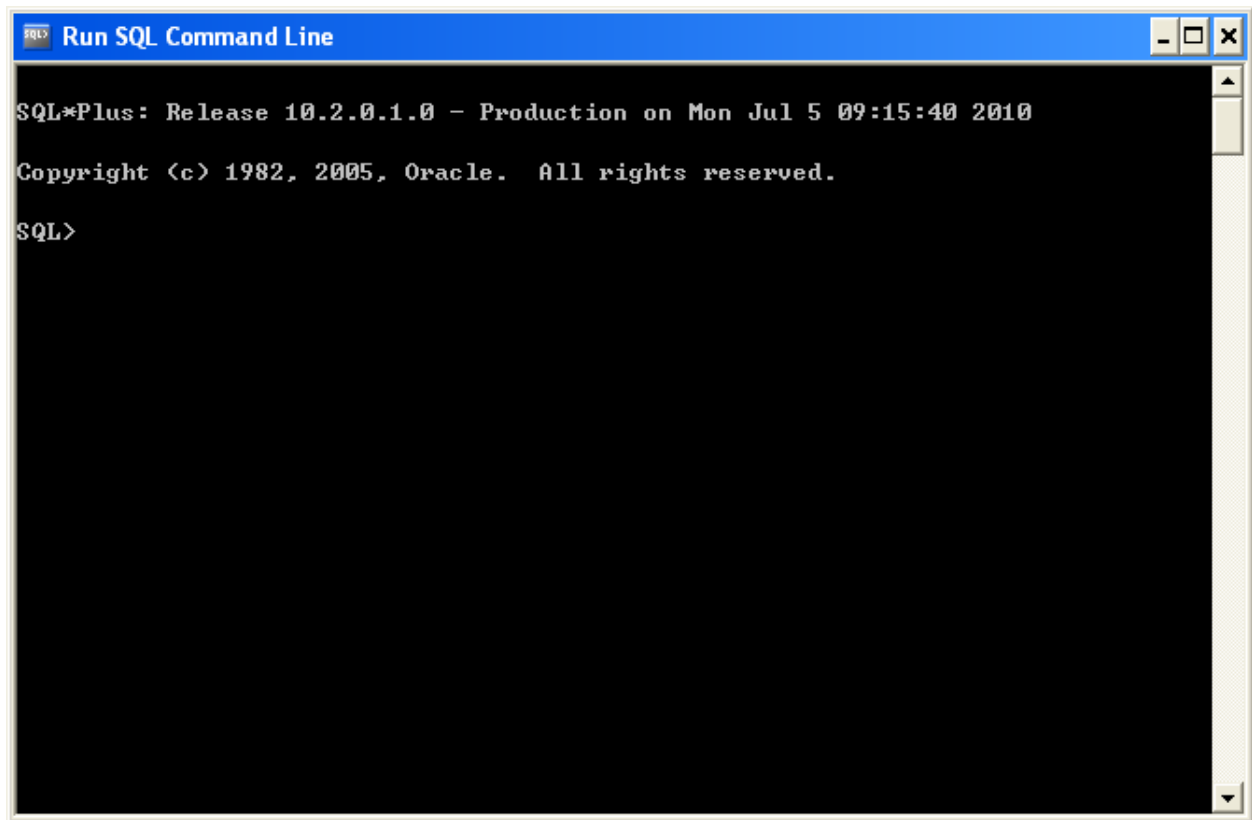
After installation, you may test to see if your Oracle XE works.

## Run SQL Plus to Test Your Oracle 10g Express Edition

Start—All Programs—Oracle Database 10g Express Edition—Run SQL Command Line, as follows.



You will then see the following screen:

A screenshot of a Windows-style window titled "Run SQL Command Line". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area is black with white text. The text reads: "SQL\*Plus: Release 10.2.0.1.0 - Production on Mon Jul 5 09:15:40 2010", "Copyright (c) 1982, 2005, Oracle. All rights reserved.", and "SQL>".

```
SQL*Plus: Release 10.2.0.1.0 - Production on Mon Jul 5 09:15:40 2010
Copyright (c) 1982, 2005, Oracle. All rights reserved.
SQL>
```

If you see the above screen shot, that means your Oracle Express Edition has been installed successfully.

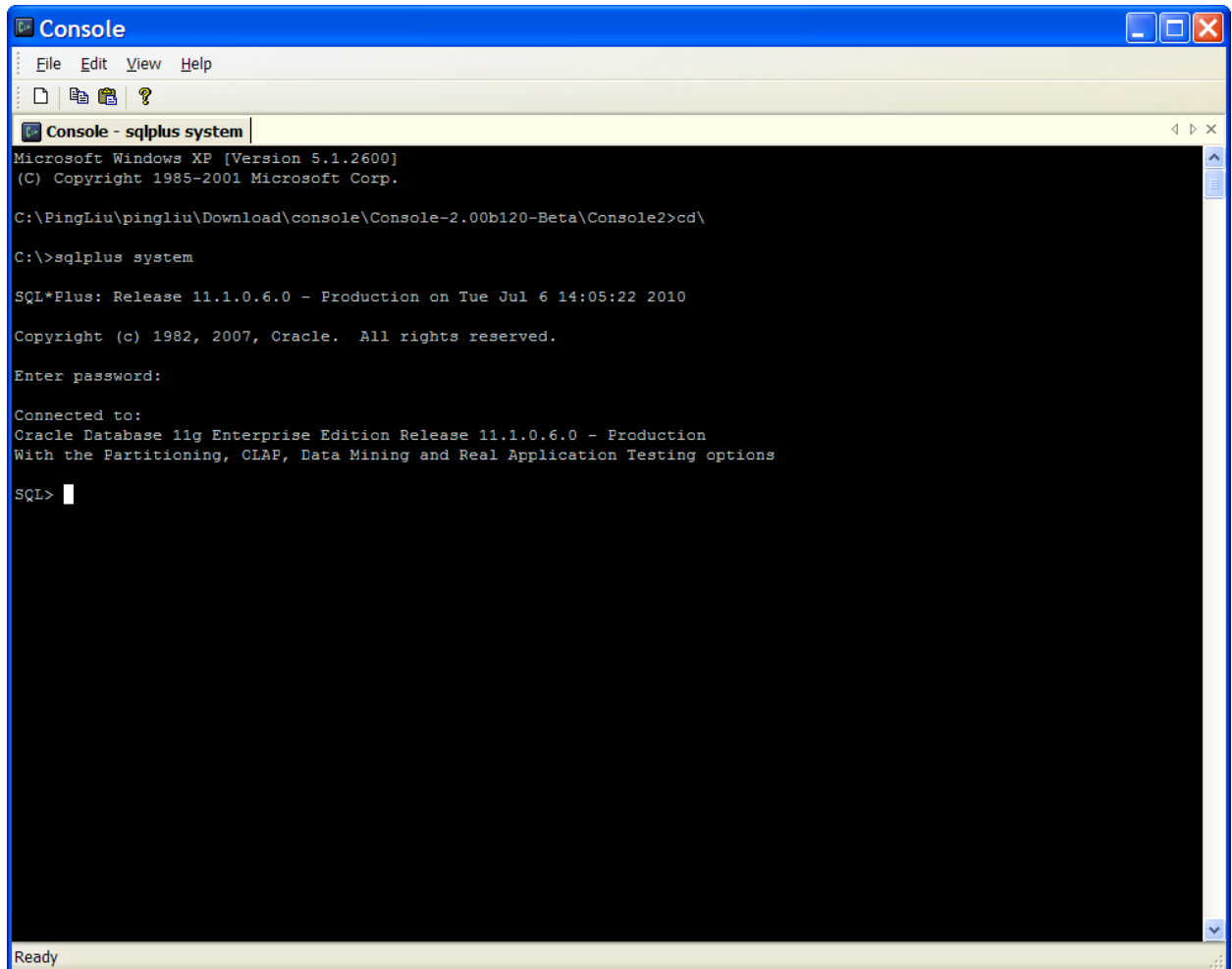
### III. Tricks and Trouble Shooting for Database Connection

You will need to do something to ensure you are able to connect to Oracle database using your PHP program. The following list those factors. If you are lucky enough, some of the issues have already been resolved automatically during installation. If you are not that lucky, you may follow those steps. It is going to be a good learning experience.

a. Oracle database has to be up and running. You cannot talk to a dead person.

The above example of running SQL Plus can be one of the ways to know if your database is running or not. If you are able to login your database using user: SYSTEM and password (as you entered during your installation), you will see something similar to the above screen shot.

Depending on your version and your operating system, the way of login your SQL Plus may be slightly different. The following shows you one example on how I login in SQL Plus in Oracle 11g (on my desk top).



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\PingLiu\pingliu\Download\console\Console-2.00b120-Beta\Console2>cd\

C:\>sqlplus system

SQL*Plus: Release 11.1.0.6.0 - Production on Tue Jul 6 14:05:22 2010

Copyright (c) 1982, 2007, Oracle. All rights reserved.

Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

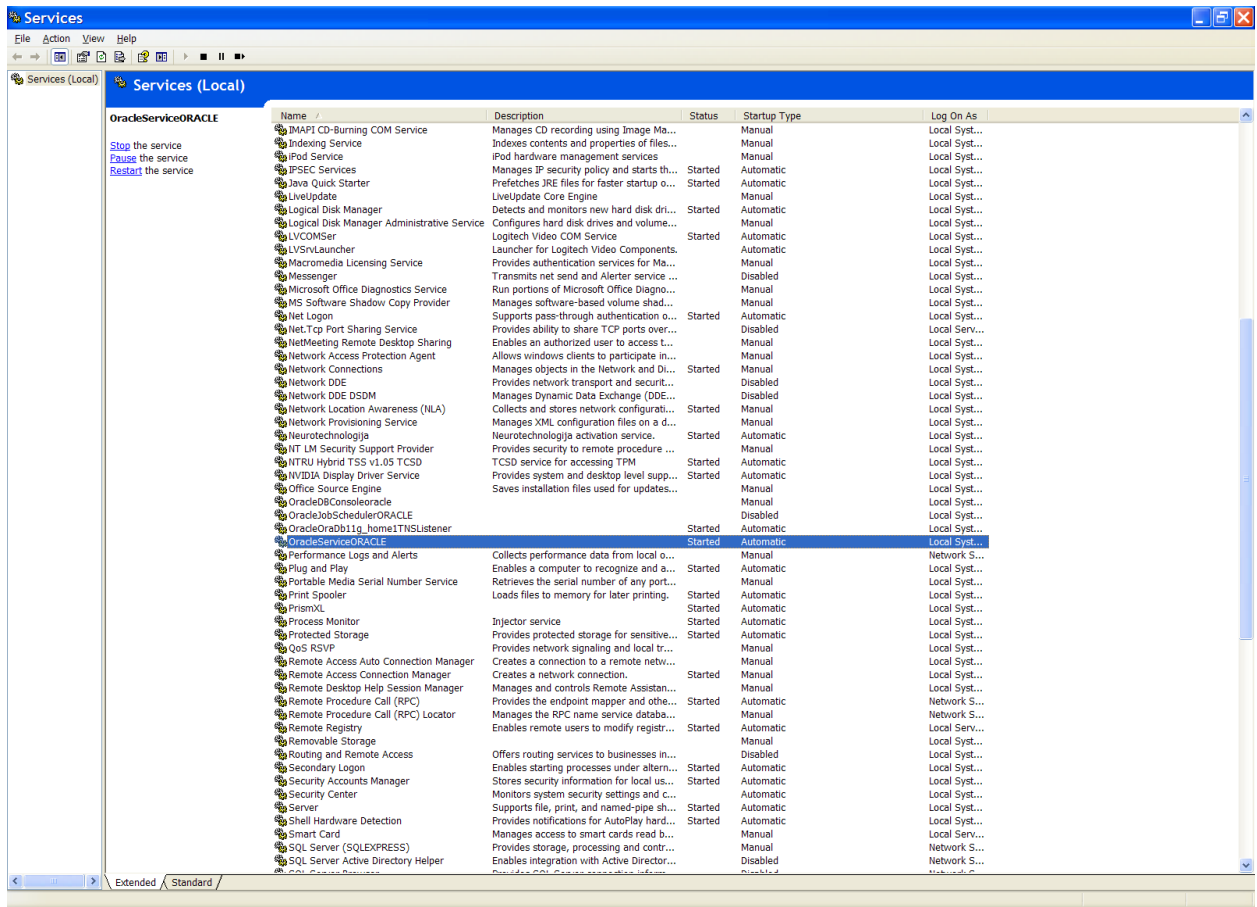
SQL>
```

The general syntax is:

```
C:\sqlplus system/password;
```

If you forget your password for the SYSTEM schema/user, you may have to call me or someone else in the class. We will help you out as long as you promise to return the favor some day.

For Windows User: Another way to know if your Database is running or not is to use Control Panel and Services, as in the following example:

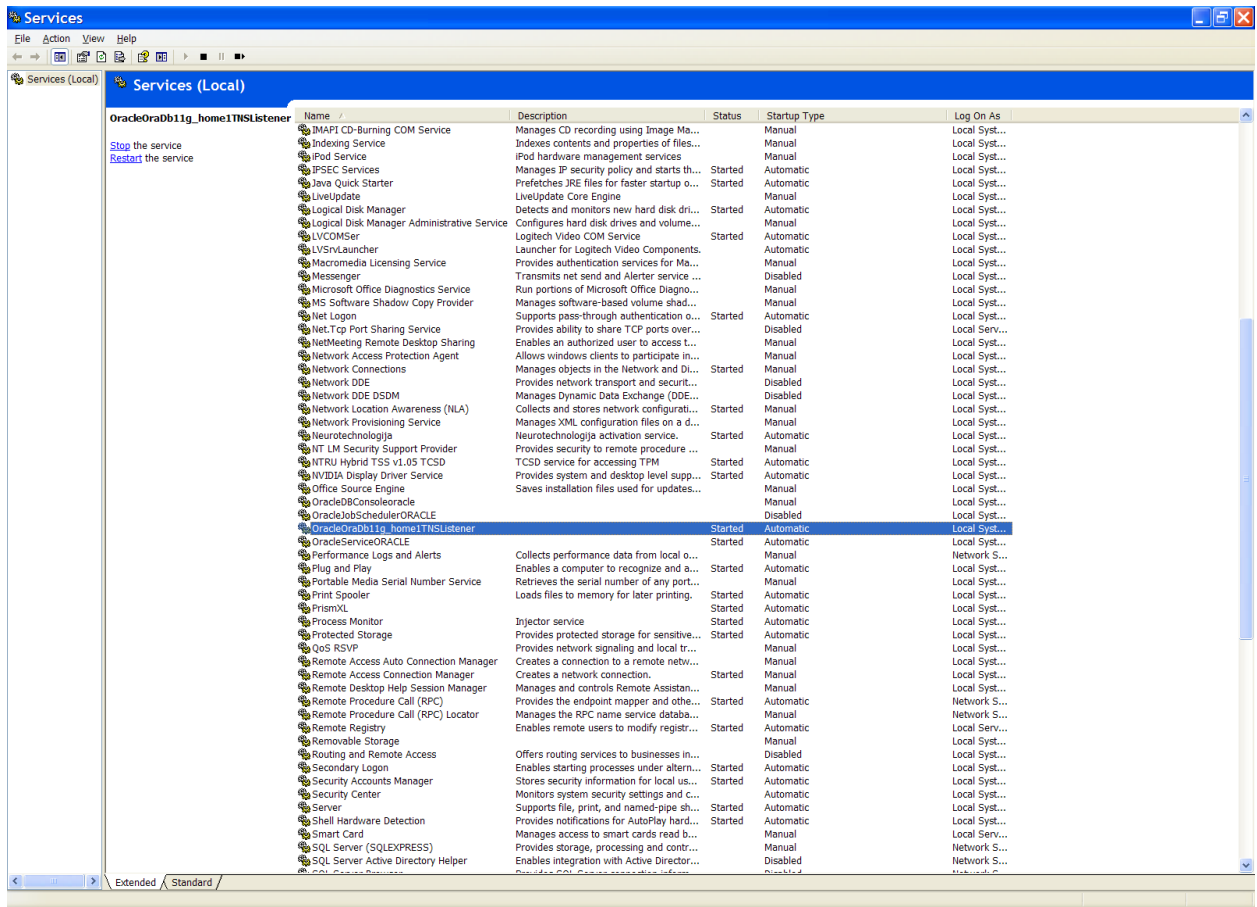


If you see “OacleServiceORACLE” is “started,” you are ready to go. Of course, depending on how you named your database during installation, your service name may be different. (I named my database instance as “ORACLE” during installation, as you can see from the above example.)

When any students approached me in the past, saying “I cannot connect to Oracle database.” or “I have error from my database.” The first question I ask is always “are you sure your database is up and running?”

b. Your TNS Listener (from Oracle) has to on: You cannot talk to a deaf.

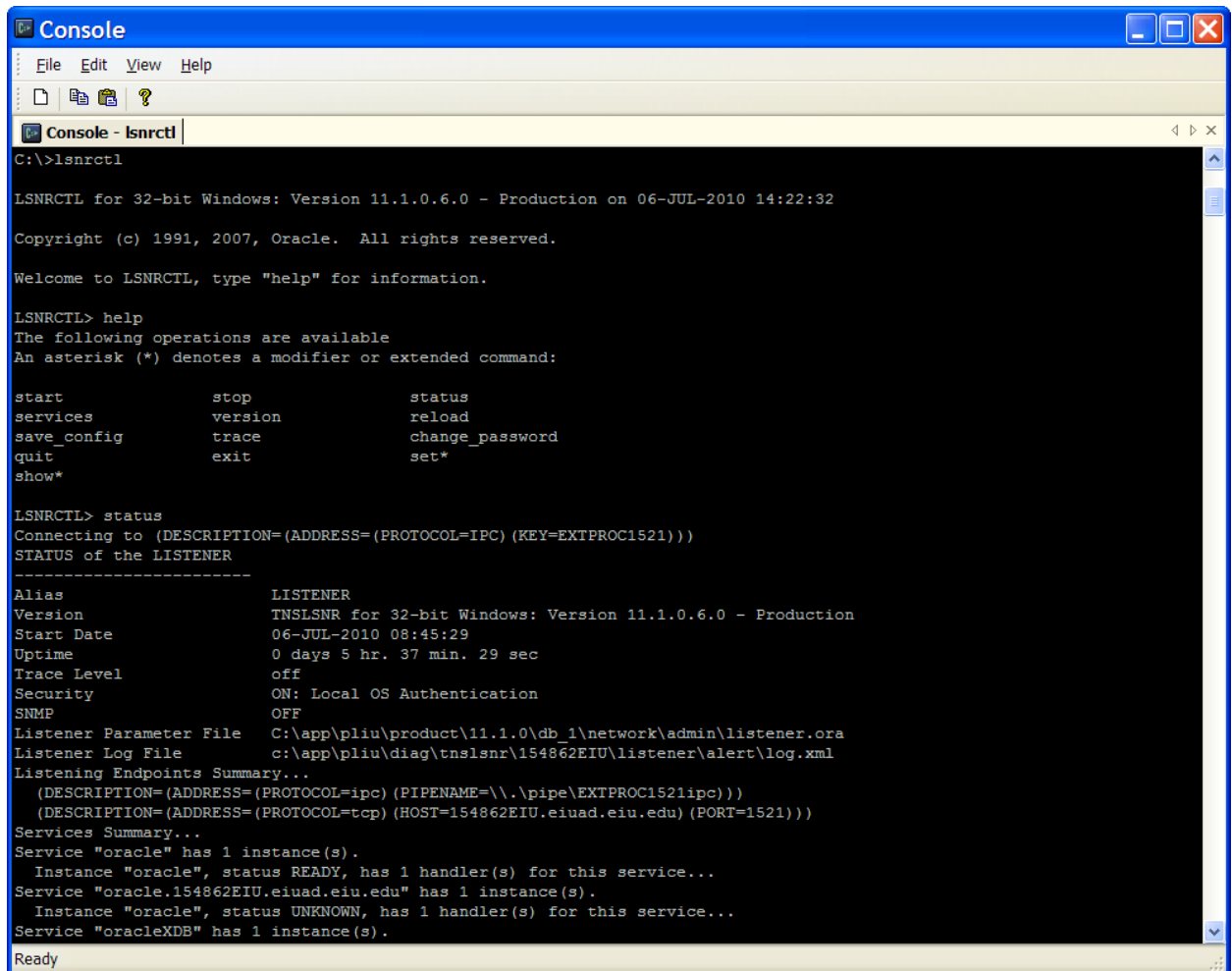
The fist way to know if the Oracle TNS Listener has been started is to go to Control Panel and Services (for Windows).



You can see from the above that my TNS listener is up and running. If your TNS Listener is not running yet, you need to trouble shoot and make sure the TNS listener is up before you can use your PHP program to test the Oracle connection.

Another way is to use the command line command from Oracle, known as "lsnrctl." The following screen shot shows you an example.





```
C:\>lsnrctl

LSNRCTL for 32-bit Windows: Version 11.1.0.6.0 - Production on 06-JUL-2010 14:22:32

Copyright (c) 1991, 2007, Oracle. All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> help
The following operations are available
An asterisk (*) denotes a modifier or extended command:

start          stop          status
services       version      reload
save_config    trace        change_password
quit           exit         set*
show*

LSNRCTL> status
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC) (KEY=EXTPROC1521)))
STATUS of the LISTENER
-----
Alias                LISTENER
Version              TNSLSNR for 32-bit Windows: Version 11.1.0.6.0 - Production
Start Date           06-JUL-2010 08:45:29
Uptime               0 days 5 hr. 37 min. 29 sec
Trace Level          off
Security             ON: Local OS Authentication
SNMP                 OFF
Listener Parameter File C:\app\pliu\product\11.1.0\db_1\network\admin\listener.ora
Listener Log File    c:\app\pliu\diag\tnslsnr\154862EIU\listener>alert\log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (PIPENAME=\\.\pipe\EXTPROC1521ipc)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=154862EIU.eiuad.eiu.edu) (PORT=1521)))
Services Summary...
Service "oracle" has 1 instance(s).
  Instance "oracle", status READY, has 1 handler(s) for this service...
Service "oracle.154862EIU.eiuad.eiu.edu" has 1 instance(s).
  Instance "oracle", status UNKNOWN, has 1 handler(s) for this service...
Service "oracleXDB" has 1 instance(s).

Ready
```

You may use:

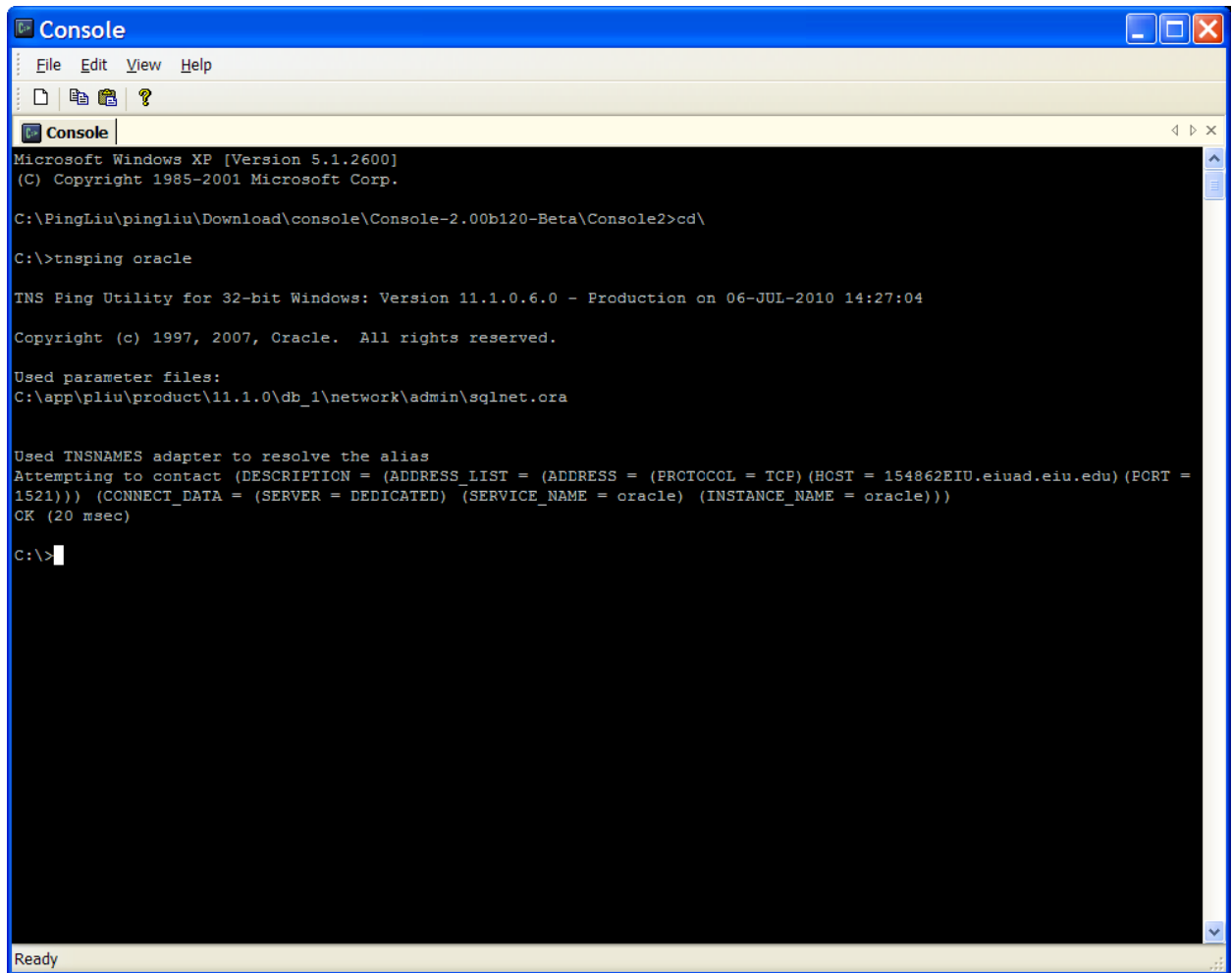
C:\>lsnrctl status

To see the status of your TNS Listener.

The "lsnrctl" is a utility that comes from Oracle. It applies to any operating system platform.

Another useful utility (from Oracle) that will be handy for trouble shooting Oracle connection is: tnsping. For example, if you have a database called "oracle," and if you want to see if and how your TNS listener communicate with your database, you may issue:

C:\>tnsping oracle



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\PingLiu\pingliu\Download\console\Console-2.00b120-Beta\Console2>cd\

C:\>tnsping oracle

TNS Ping Utility for 32-bit Windows: Version 11.1.0.6.0 - Production on 06-JUL-2010 14:27:04

Copyright (c) 1997, 2007, Oracle. All rights reserved.

Used parameter files:
C:\app\pliu\product\11.1.0\db_1\network\admin\sqlnet.ora

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = 154862EIU.eiuad.eiu.edu)(PORT = 1521))) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = oracle) (INSTANCE_NAME = oracle)))
OK (20 msec)

C:\>
```

The above “Isnrctl” and “tnsping” utilities will give you plenty of information to trouble shoot your Oracle connection.

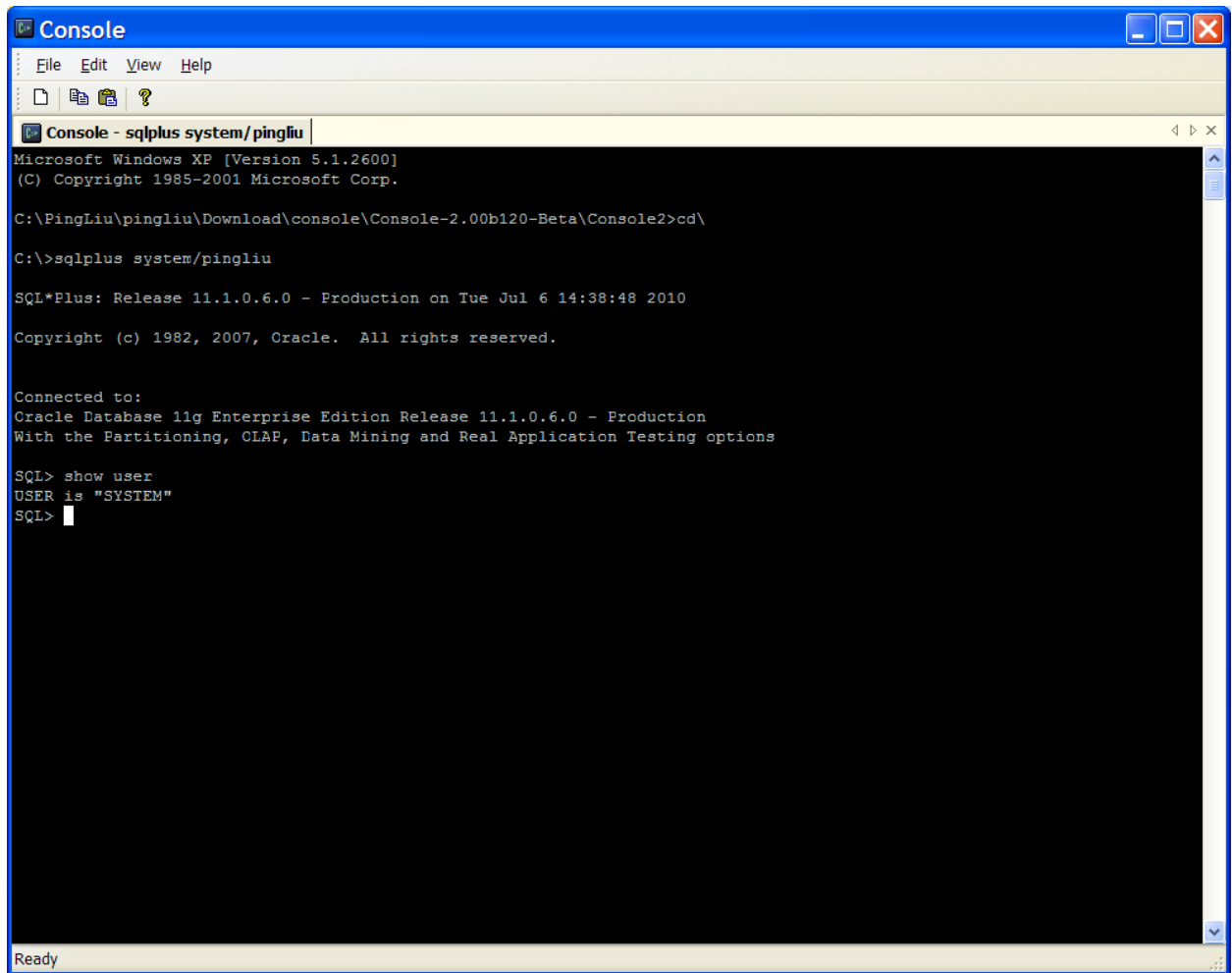
The trouble shooting of TNS listener will require some extensive experience, especially if you had problems during installation. Contact me If you could not resolve the TNS listener issue.

c. You have to have something in your database so that you may ask Oracle database.

To do that, you need to create an Oracle account, as follows (for Windows):

```
C:\sqlplus system/password
```

Once you are able to login Oracle SQL Plus, you will see something similar to the following:

A screenshot of a Windows console window titled "Console". The window has a menu bar with "File", "Edit", "View", and "Help". Below the menu bar is a toolbar with icons for file operations. The main area of the console shows the following text:

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\PingLiu\pingliu\Download\console\Console-2.00b120-Beta\Console2>cd\

C:\>sqlplus system/pingliu

SQL*Plus: Release 11.1.0.6.0 - Production on Tue Jul 6 14:38:48 2010

Copyright (c) 1982, 2007, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> show user
USER is "SYSTEM"
SQL>
```

The status bar at the bottom of the window shows "Ready".

Here I used a SQL command “show user” to confirm that I have logged in as “SYSTEM.” The “SYSTEM” user will allow you to create other users such as “php.”

Now, you are ready to create a user called “php” with a password “php”, as in your textbook, on page 44.

```
SQL> CREATE USER php IDENTIFIED BY php;
```

You may then grant some necessary privileges to the user “php” so that “php” can login the database.

```
SQL> GRANT resource, connect TO php;
```

Once the above grant is successful, you should be able to login as “php” with a password “php.”

```
SQL>connect php/php;
```

```
Console
File Edit View Help
Console - sqlplus system/pingliu
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\PingLiu\pingliu\Download\console\Console-2.00b120-Beta\Console2>cd\

C:\>sqlplus system/pingliu

SQL*Plus: Release 11.1.0.6.0 - Production on Tue Jul 6 14:38:48 2010

Copyright (c) 1982, 2007, Oracle. All rights reserved.

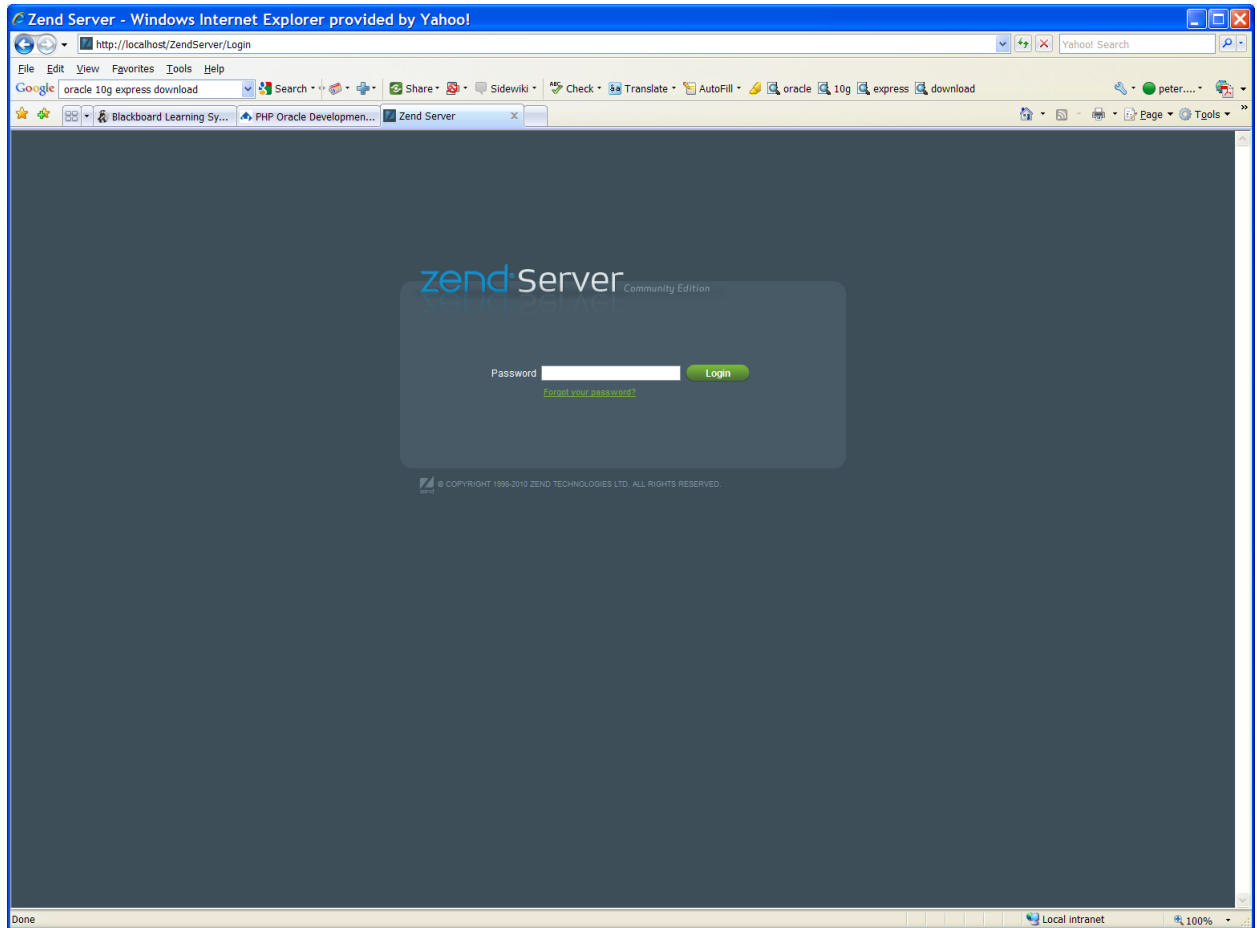
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> show user
USER is "SYSTEM"
SQL> connect php/php;
Connected.
SQL>
```

The user (“php”) and password (“php”) will be used now to test if your PHP program can talk to the Oracle database.

d. The above tasks are meant to make sure your Oracle database is alive before anyone (including your PHP program) can communicate with the database. Now, we need to make sure that the PHP engine understands the communication protocol or language. The communication between PHP and Oracle is handled by OCI8 (Oracle Call Interface 8). OCI8 is your agent who negotiates between Oracle and PHP.

OCI8 came with your Zend Server as you have already installed earlier this semester. But, you need to make sure it is on in your PHP engine. To do it, please login your Zend Server, as follows:

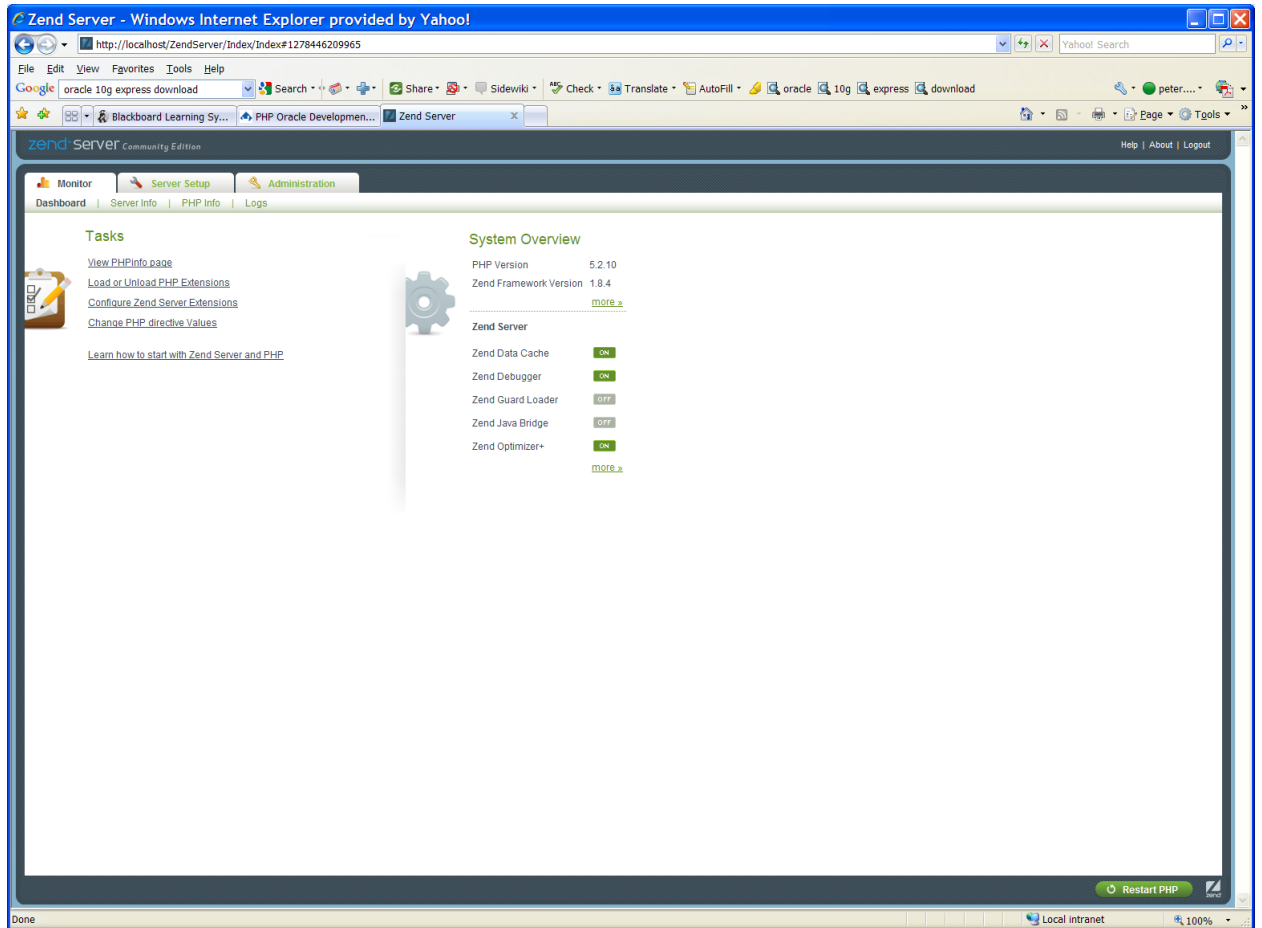


Please note: The following example is for Zend Server Community Edition. As of this writing on June 5, 2015, Zend offers a Developer Edition. It may vary slightly from the following procedure. But, the idea remain similar. Use it as your reference.

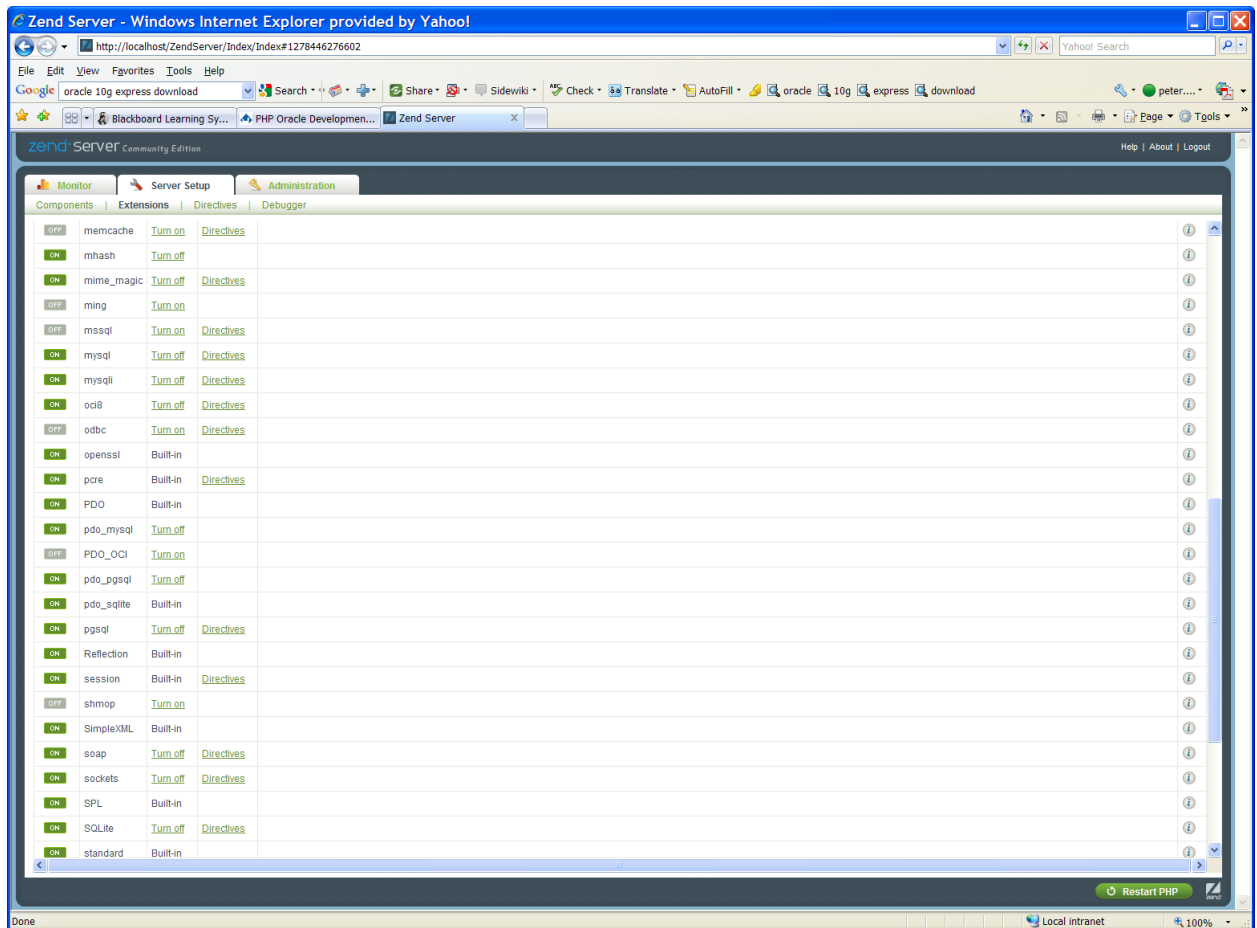
If your installation is similar to mine, you may access the Zend Server with the following link:

<http://localhost/ZendServer/Login>

Once you login the Zend Server, you will see something like:



You can then drill to Server Setup, Extensions, and find OCI8, as follows:



Turn on OCI8, and “Restart PHP.” Your communication agent is awake and working for you.

e. Your parameters in PHP have to be correct. You have to speak to the same language (English, Chinese, Hindi or others) as you talk to others.

I am going to ask you to use the [OracleStandardConnection.php](#) (pages 351-352) to test if you are able to connect to your Oracle database. To prepare you for more complex projects in the future, however, I am going to ask you to slightly modify the above code.

```
$schema='php'; //The schema defines the user name you like to access in Oracle database
$password='php'; //It defines the password of the above schema/user
$dbdatabase='xe'; //It defines the database (instance) name
```

Those are the three parameters that you will use to communicate with Oracle database. Your [OracleStandardConnection.php](#) code can be modified as:

```

<?php
$schema='php'; //The schema defines the user name you like to access in Oracle database
$password='php'; //It defines the password of the above schema/user
$database $database='xe'; //It defines the database (instance) name

if ($c=@oci_connect($schema,$password,$database))
{
Print "Successfully connected to Oracle database.<br>";

oci_close($c);
}
else
{
$errorMessage=oci_error();

print '<table border="1" cellpadding="0" cellspacing="0">';

foreach ($errorMessage as $name=>$text)
{
print '<tr><td>'.$name.'</td>';

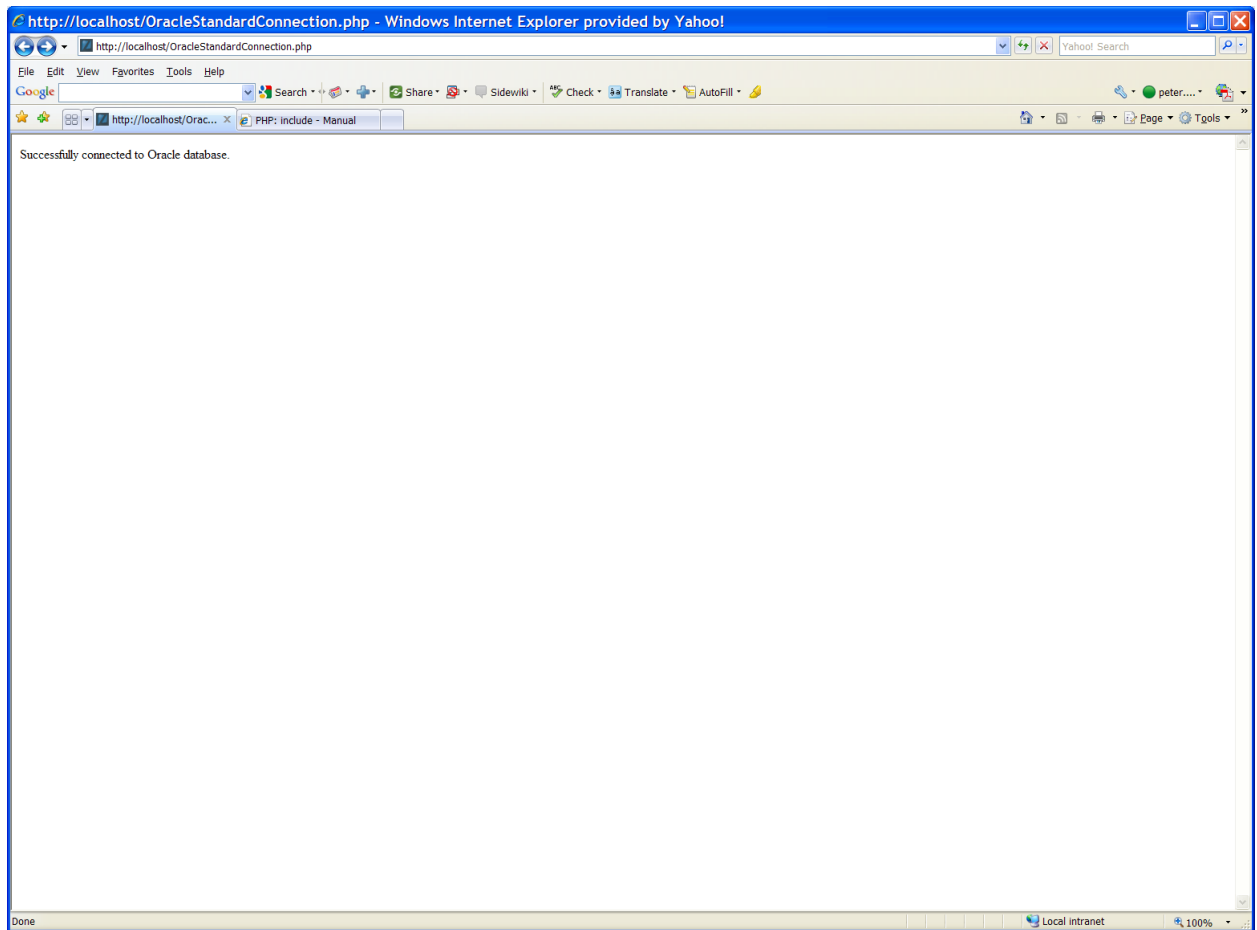
print '<td>'.$text.'</td></tr>';
}

print '</table>';
}
?>

```

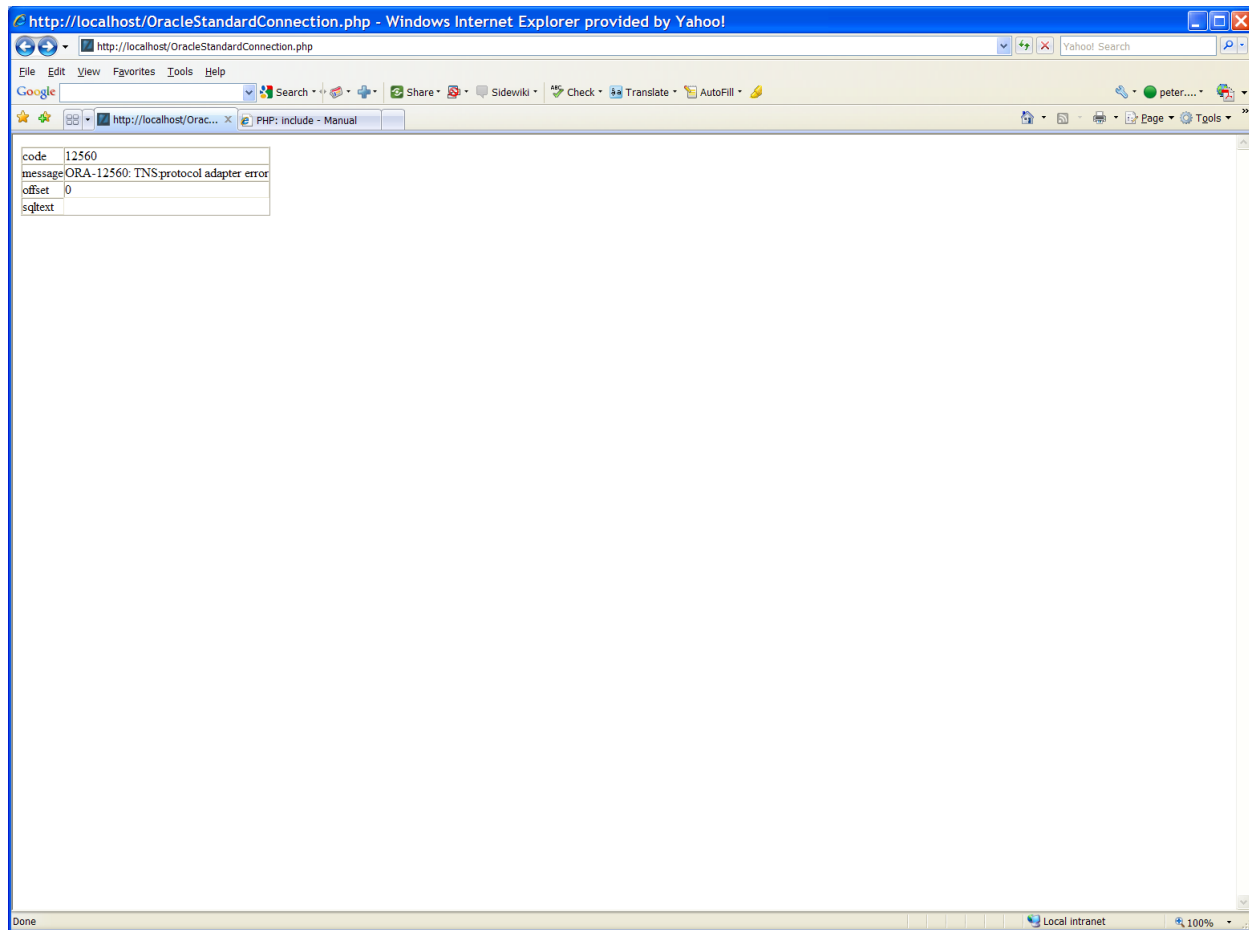
If you installed Oracle 10g Express Edition, chances are the above code may produce the result you like, as in the following figure.





If you are not lucky enough yet, you may have to do some more work, before you can declare a victory. Here are the possibilities, assuming that you have followed all the steps above:

- a. If you have installed Oracle 10g, 11g Express Edition, the code still gives you error message such as the following:



You may redefine your \$database variable as:

```
$database='localhost/xes';
```

That was the case on my laptop.

- b. If you installed Oracle 9i, 10g, 11g or 12c Enterprise Edition, the most reliable way is to use "tnsping" utility to get the database connection string, as follows:

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\PingLiu\pingliu\Download\console\Console-2.00b120-Beta\Console2>cd\

C:\>tnsping oracle

TNS Ping Utility for 32-bit Windows: Version 11.1.0.6.0 - Production on 06-JUL-2010 16:00:03

Copyright (c) 1997, 2007, Oracle. All rights reserved.

Used parameter files:
C:\app\pliu\product\11.1.0\db_1\network\admin\sqlnet.ora

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = 154862EIU.eiuad.eiu.edu)(PORT = 1521))) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = oracle) (INSTANCE_NAME = oracle)))
OK (20 msec)

C:\>
```

If you see your tnsping is successful, that means that all communication channel to Oracle database base is clear. The “OK (20 msec)” is a good news for you. Then, you may use the database string to redefine your \$database variable, as:

```
$database=' (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = 154862EIU.eiuad.eiu.edu)(PORT =1521))) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = oracle) (INSTANCE_NAME = oracle)))';
```

If you follow the above steps, the above approach will never fail. Good luck.

**Challenge:** There are other ways to find out the database connection string above. One way is to look for the string in tnsnames.ora file. I hope you can find the file in your computer. Then, you can copy the connection string. Just do not change the file.

#### IV. To Prepare for the Future

Now, you have made the connection with Oracle database successfully. Your PHP code will look like the following if you have installed Oracle Enterprise Edition.

```
<?php

$schema='php'; //The schema defines the user name you like to access in Oracle database
$password='php'; //It defines the password of the above schema/user
$database="(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST =
154862EIU.eiuad.eiu.edu)(PORT =1521))) (CONNECT_DATA = (SERVER = DEDICATED)
(SERVICE_NAME = oracle) (INSTANCE_NAME = oracle))";

//It defines the database (instance) name
if ($c=@oci_connect($schema,$password,$database))
{
Print "Successfully connected to Oracle database.<br>";
oci_close($c);
}

else

{
$errorMessage=oci_error();
print '<table border="1" cellpadding="0" cellspacing="0">';
foreach ($errorMessage as $name=>$text)
{
print '<tr><td>'.$name.'</td>';
print '<td>'.$text.'</td></tr>';
}
print '</table>';
}
?>
```

This seems to be very easy and straight forward. Nonetheless, what if you have 200 PHP programs that need to connect to Oracle database, how can you do it (to define the database connection parameters)? Of course, at the beginning, you could elect to copy and

paste the parameters to those 200 PHP files. That does not seem to be too difficult, either. You can do it with a little sweat.

What if after two years, a new database administrator in your organization decides to upgrade your system. As a result of the upgrade, the database is named as "oracle1" from the old name "oracle." What do you do with those 200 PHP files with such a slight change? Chances are if I do not document them well, I will not be able to find those 200 files easily. Of course, your new system will end up with a lot of problems after the upgrade. Your customers including your boss and your wife/husband will not be happy. This is a typical issue of "scalability."

In order to make your PHP/Oracle system easily scalable in the future, you can define your Oracle database parameters in a single file. For the purpose, I am going to ask you to name this file as "oracle\_parameters.php".

Now, what I am asking you to do is to move the three lines of Oracle connection parameters into a separate PHP file, named as "oracle\_parameters.php," and place this file in the same document root as your main PHP program. Your OracleStandardConnection.php code will look like as follows:

```
<?php
include_once("oracle_parameters.php");
if ($c=@oci_connect($schema,$password,$database))
{
Print "Successfully connected to Oracle database.<br>";
oci_close($c);
}
else
{
$errorMessage=oci_error();
print '<table border="1" cellpadding="0" cellspacing="0">';
foreach ($errorMessage as $name=>$text)
{
print '<tr><td>'.$name.'</td>';
```

```
print '<td>'.$text.'</td></tr>';  
  
}  
print '</table>';  
  
}  
?>
```

Now, as a project for this week, you will be creating the parameter file (oracle\_parameters.php) based upon your own computer system, and make the above code work for you.

I have tested on my computer system. It is now your turn. Have fun.