Class Note 3
PHP Oracle Development
(Updated 6/3/2015)

Control Structure

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.

As a step further, we will focus on the control structure, which is essential and common to any programming language. Control structure enables us to solve problems more efficiently, and in many cases, to help us solve complex problems.

Control structure can be divided into two (2) major categories. The first one is also known as “conditional structure” whereas the second one is referred to “iterative control structure.”

A. Conditional Structure:

Conditional structure includes two (2) types, as follows:

1. If statement

If statement can be in the following format:

```
if (condition is true)
{
Do something;
}
```

Or

```
if (condition is true)
{
Do something;
}
else
{
Do something else;
}
```

Or, in a more elaborated form

```
if (condition is true)
{
Do something;
}
else if (another condition is true)
{
Do something else;
}
else
{
Do something;
}
```

Be careful about the action involved under “else.” Many new programmers see surprises that it involves more than we thought.

We can use the above format, simple or complex, to direct the PHP to take whatever action we need.

2. Switch statement

Please read pages 90-93. Note the following:

a. You need “break;” within each case.
You will see the problem (on the Review Questions) if you miss the “break.” It may or may not give you a syntax error. Test the following code, and you will see what I mean. If you are lucky enough, the same question may appear in your mid-term test.

```
<?php

$i="bar";

switch ($i)

{

case "apple":

echo "i is apple<br>";

break;
case "bar":
echo "i is bar<br>";

case "cake":
echo "i is cake<br>";
break;

default:
echo "i is nothing<br>";
break;
}
?>
```

b. Be careful and mindful on “default” condition. Most likely, you will not see any syntax error, but your results may be illogical. This is the best time when an instructor may catch you copying code from others.

B. Iterative Control Structure:

1. While loop:

The while loop test the condition before it enters the statement.

```
<?php
$i = 1;
while ($i <= 10) {
   echo $i++."<br>";  /* the printed value would be
```

```
            $i before the increment

            (post-increment) */

//The details of the ++ operator is available at page 65 in your textbook.

//This is a very useful operator and you will see it everywhere.

//Test it if you replace it with "++$i".

}
```

Pay attention to the “++” (increment), which is very useful in practical programming.

What will happen if you write the following while loop?

```
<?php

$i = 1;

while ($i <= 10) {

  echo $i."<br>";

 }
```

Why?

2. Do-While Loop: The do-while loop does not test the condition at the beginning. It tests the condition after executing the statement(s). In other words, the statements will be executed at least once. Please study the following code.

```
<?php

$i = 2;

do {

   echo "The value of \$i is ".$i."<br>";
```

```
            echo "It displays the value even though the while condition stated that \$i has to be larger than 3.<br>";

} while ($i > 3);

?>
```

If you change the above “while ($i>3);” to “while ($i>1);” it will crash your program. Why?

If you still do not believe me, test it. You will see what I meant.

3. For Loop: Let us look at the following simplest code on a “for loop.”

```
<?php

for ($i = 1; $i <= 5; $i++) {

   print $i;

}

?>
```

where “$i=1” sets the lower boundary and “$i<=5” sets the upper boundary for the looping operation, and “$i++” will increment the $i after each round of execution.

What will be the output from the above code? (This is a test question.)

What will happen if you have the following code?

```
<?php

for ($i = 5; $i >= 1; $i--) {

   print $i;

}

?>
```

4. Foreach Loop: This is the most useful control structure while you want to display data from a database table. You need the skill to be an effective programmer on database-driven web pages.

   Foreach loop is often associated with array application. We will deal with arrays on the next chapter. In the meantime, you have to accept what you have seen on the array. It works well.

```
<?php
$myArr = array("one", "two", "three");
foreach $myArr as $value) {
   echo "Value: $value<br />\n";  //<br /> is just a line break.
}
?>
```

   The code will automatically set the looping condition until the array is exhausted. Test the code. Once you are familiar with concept, you will like it unless you like to work hard for nothing.

   Because of its importance, we will do a project using the “foreach” loop. It is extremely useful while dealing with database table applications.