SELECTION AND REPETITION IN PROGRAMMING

A. Selection in Programming

The ‘if’ Statement

The ***if*** statement allows you to examine data and take actions based on its value. The structure is as follows:

```
if(condition)
{
      code block;
}
```

If the condition is true then the code in the code block that follows will be executed, otherwise it will be skipped.
It is standard practice to indent the code that is inside the curly braces. For example:

```
if ($a > $b)
{
  print “a is bigger than b”;
}
```

Exercise 5

1. Design a web page which will contain the PHP code to store the value 7 in a variable called $num.
   a) If the variable is > 0, then display in the browser “The number is greater than 0”.
   b) If the variable is equal to 7, then display in the browser “The number is 7”.
   c) Save the file as **ex5-1.php**

2. Change the number to –7 and open the page in the browser again. (Nothing should be displayed).
   Save the file as **ex5-2.php**

The ‘else’ Statement

It is customary to use the ***else*** statement so that its code block is executed if the condition is false.
The structure is as follows:

```
if(condition)
{
      code block
}
else
{
      alternate code block
}
```

It is often necessary to execute a statement if a certain condition is met, and execute a different statement if the condition is not met. This is what the “else” does. ‘else’ extends an if statement to execute a separate statement in case the expression in the if statement evaluates to false. The else statement is executed only after the if expression has been evaluated to false. For example:

```
if ($a > $b)
{
   print “a is bigger than b”;
}
else
{
   print “a is not bigger than b”;
}
```

Exercise 5 cont.

3. Design a web page which will contain the PHP code to store the value 7 in a variable called $num. If the variable is > 0 then display in the browser “The number is greater than 0”, if the variable is <= 0 display “The number is less than or equal to 0”. Finally, display “The number is $num”

4. Change the code so that the number is equal to –7 the second time.

5. Design a web page which will contain the PHP code to store the value ‘blue” in a variable called $colour. If the variable is blue then display “My favourite colour is blue”, otherwise display “My favourite colour is not $colour”.

6. Change the code so that the $colour = ‘red’.

The ‘elseif’ Statement

The elseif statement is executed after the initial if expression has been evaluated to FALSE. There is no limit to the number of elseif conditions that can be evaluated within an if expression. After every elseif statement has been evaluated to FALSE, the else statement is evaluated. The syntax is:

```
if (expression)
{
   statements
}
elseif (expression2)
{
   statements
}
elseif (expression3)
{
   statements
}
else
{
   statements
}
```

For example

```
if ($a > $b)
{
   echo “a is bigger than b”;
}
elseif ($b > $a)
{
   echo “b is bigger than a”;
}
else
{
   echo “a and b are the same”;
}
```

Exercise 6

1. Store a simple name in a variable called $name. If the name = “Jane” print “Hello Jane”, if the name equals “Sally” print “Hello Sally”, if the name equals “Karen” print “Hello Karen” if the name equals any other name print “Hello new person”.

2. Design a web page which will contain the PHP code to store the value 7 in a variable called $num. If the variable is > 0 then display in the browser “The number 7 is greater than 0”, if the variable is < 0 display “The number is less than 0”. If the variable is equal to 0 display “The variable is equal to 0”. Change the code so that the number is equal to –7 the second time and 0 the third time.

The ‘switch’ Statement (case)

The ***switch*** statement works much like the *if* statement. It is useful when multiple choice is required. The structure is as follows:

```
switch(variable name)
{
      case value1:
            statements;
            break;

      case value2:
            statements;
            break;

      case value3:
            statements;
            break;

      default:
            statements;
            break;
}
```

The switch statement executes one statement at a time. PHP continues to execute the statements until the end of the switch block “}”, or until the ***break*** statement is encountered. If a break statement is not contained at the end of each case statement list, PHP continues executing all subsequent case statements. A special case is the ***default*** case. This case matches anything that wasn’t matched by the other cases. For example:

```
switch ($i)
{
   case 5:
      echo ‘$i equals 5’;
      break;
   case 12:
      echo ’$i equals 12’;
      break;
   case 2:
      echo ‘$i equals 2’;
      break;
   default:
      echo ‘$i is not equal to 5, 12 or 2’;
}
```

Exercise 6 cont.

3. Create a web page which will contain the PHP code to store the value ‘blue” in a variable called $colour. If the variable is blue then display “The colour is blue”, if colour is red display “The colour is red”, if colour is yellow display “The colour is yellow”. If it is another colour then display “The colour is not blue, red nor yellow”.

4. Use ex6-1.php and change the code to incorporate ‘switch’. Save the file as **ex6-4.php**

B. Repetition in Programming

We use **loops** to repeat the same task, or, use loops to automate repetitive processes. We will be looking at the ***'for'*** and ***'while'*** loops.

The 'for' Statement

for is a loop statement and its structure looks like:

```
for(initialisation;  test;  increment)
{
  statement
}
```

where:

initialisation	usually starts a loop index, or loop counter	
test	is a conditional statement involving the loop index which permits the loop to continue while true	
increment	a statement executed after each loop (usually used to increment or decrement the loop index)	
statement	can be	i) one statement or ii) compound statements which is a code block surrounded by { } and is made up of many lines of code

For example:

```
for ($i=1; $i<10; $i++)
{
        print $i;         //$i is the loop index
}
```

Exercise 7

1. Use the ***for*** loop to display all numbers from 1 to 10, in a column. Save file as **ex7-1.php**
2. Use the ***for*** loop to display all numbers from 10 to 1, in a column
3. Start a new Web page that will display all odd numbers from 1 to 15. Create PHP script so that you use the ***for*** statement.
4. This Web page will display the squares of numbers from 1 to 5 using the ***for*** loop.

The 'while' Statement

The ***'while'*** is a loop statement and its structure looks like:

```
while (expression)
{
  statement;
}
```

The loop keeps executing its statement while the expression is true. Unlike the *for* loop, the *while* loop does not require a loop index and you can use any expression that evaluates to a true or false.

For example

```
$i=1;
while ($i<10)
{
        print $i;
        $i++;
}
```

Exercise 8

1. Use the ***while*** loop to display all numbers from 1 to 10, in a column. Save file as **ex8-1.php**

2. Use the ***while*** loop to display all odd numbers between 1 and 15, in a column.

3. Use the ***while*** loop to display the squares of numbers between 1 and 5, in a column.

The 'do while' Statement

The do while loop is similar to the while loop except that the condition is checked at the bottom of the loop, insuring that all statements contained between the "{ }" are evaluated at least once.

```
do
{
     statements;
}while(condition);
```

For example:

```
$i=0;
do
{
     print $i;
     $i++;
}while($i<10);
```

Exercise 8 cont.

4. Using a do while loop display all numbers divisible by 3 starting at 3 up to 30. Save file as **ex8-4.php**

The 'foreach' loop

The 'foreach' loop works only on arrays. The syntax is as follows:

```
$months = array("January, "February", "March", "April");
foreach($months as $value)
{
  echo $value . ", ";
}
//will display January, February, March, April,
```

You can display the key (index number) as well as the value of the array elements. Syntax is:

```
$months = array("January, "February", "March", "April");
foreach($months as $key => $value)
{
  echo "Key: $key;  Month: $value <br />";
}
//will display Key: 0; Month: January
//             Key: 1; Month: February
//             Key: 2; Month: March
//             Key: 3; Month: April
```

Exercise 9

1. Create a php file that will store your 4 favorite ice-cream flavours in an array and it will display these flavours in a line; for example "My favorite ice-cream flavours are – chocolate, berry, mocha, mango".

2. Modify file ex9-1.php, so that the flavours are displayed in a column with the keys next to them. The key number should start with a '1' and not with a '0'. Save file as **ex9-2.php**

3. Create a web page that will store three names and their ages in an associative array. It will then display the three names and their respective ages in a column.