

PHP

Part 2

Topics

- PHP Conditional Statements
- PHP Loops
- PHP Functions
- PHP Arrays
- PHP Global Variables

PHP Conditional Statements

- PHP we have the following conditional statements:
- **if statement** - executes some code only if a specified condition is true
- **if...else statement** - executes some code if a condition is true and another code if the condition is false
- **if...elseif....else statement** - selects one of several blocks of code to be executed
- **switch statement** - selects one of many blocks of code to be executed

PHP - The if Statement

```
if (condition) {  
    code to be executed if condition is true;  
}
```

PHP - The if Statement

Example

```
<?php
$t=date("H");

if ($t<"20") {
    echo "Have a good day!";
}
?>
```

PHP - The if...else Statement

```
if (condition) {  
    code to be executed if condition is true;  
} else {  
    code to be executed if condition is false;  
}
```

PHP - The if...else Statement

Example

```
<?php
$t=date("H");

if ($t<"20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

PHP - The if...elseif....else Statement

```
if (condition) {  
    code to be executed if condition is true;  
} elseif (condition) {  
    code to be executed if condition is true;  
} else {  
    code to be executed if condition is false;  
}
```

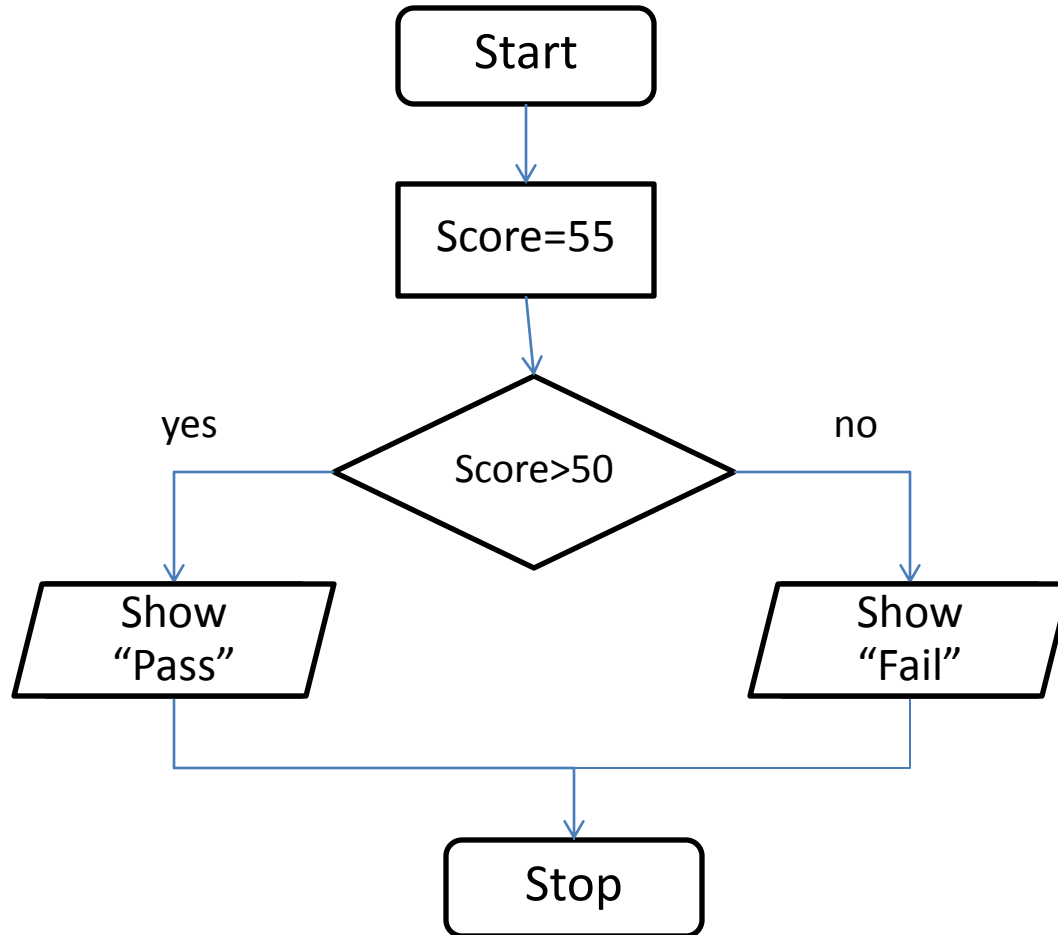

PHP - The if...elseif....else Statement

Example

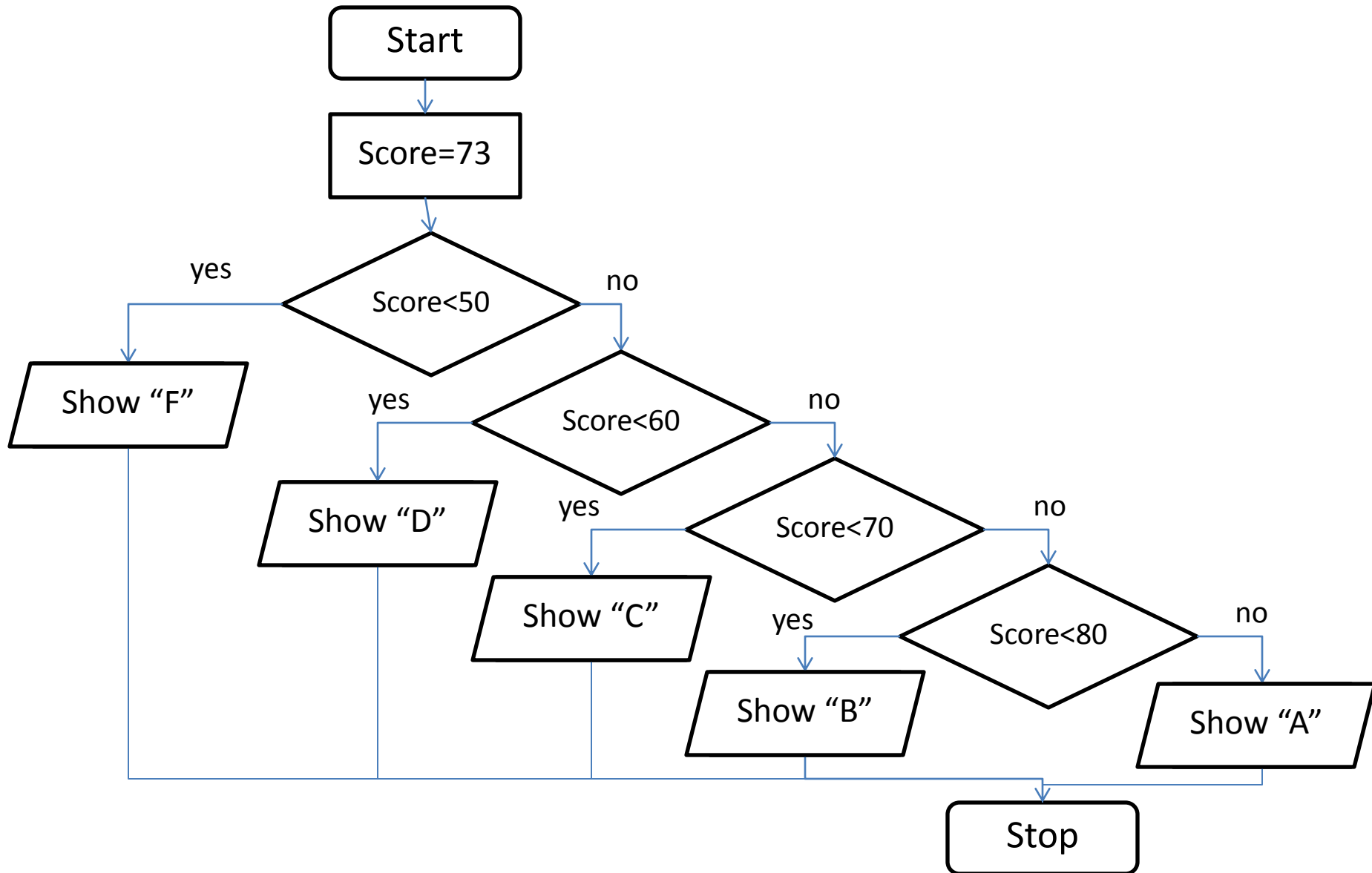
```
<?php
$t=date("H");

if ($t<"10") {
    echo "Have a good morning!";
} elseif ($t<"20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

if Statement LAB#1



if Statement LAB#2



The PHP switch Statement

```
switch (n) {  
    case label1:  
        code to be executed if n=label1;  
        break;  
    case label2:  
        code to be executed if n=label2;  
        break;  
    case label3:  
        code to be executed if n=label3;  
        break;  
    ...  
    default:  
        code to be executed if n is different from all labels;  
}
```

The PHP switch Statement

Example

```
<?php
$favcolor="red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, or green!";
}
?>
```

The PHP switch Statement

```
<?php
$t=date("H");

switch($t){
    case 8:
        echo "กินข้าวเช้า";
        break;
    case 12:
        echo "กินข้าวเที่ยง";
        break;
    case 18:
        echo "กินข้าวเย็น";
        break;
    default:
        echo "ไม่กินข้าว";
}

?>
```

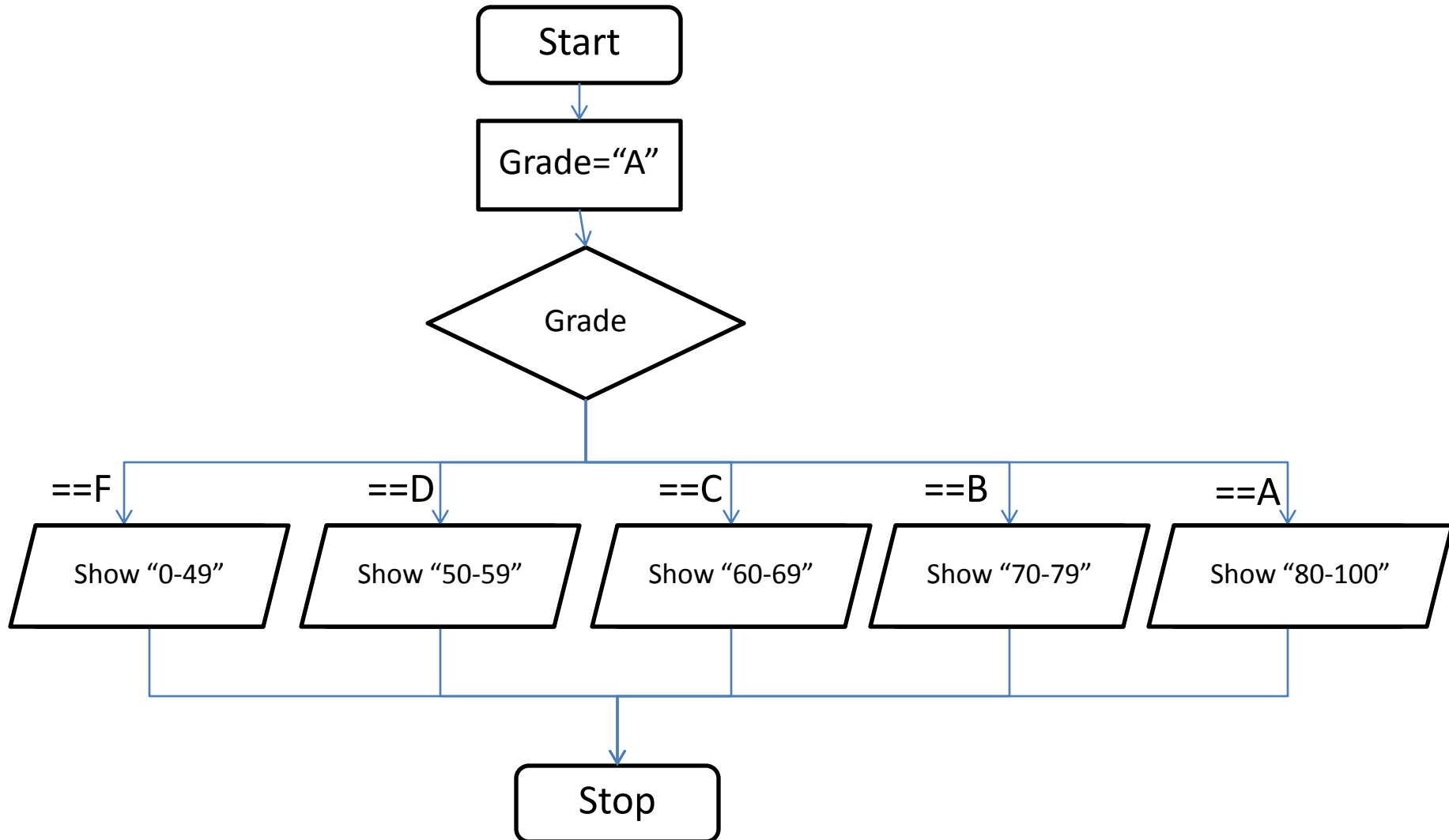
The PHP switch Statement

```
<?php
$t=date("H");

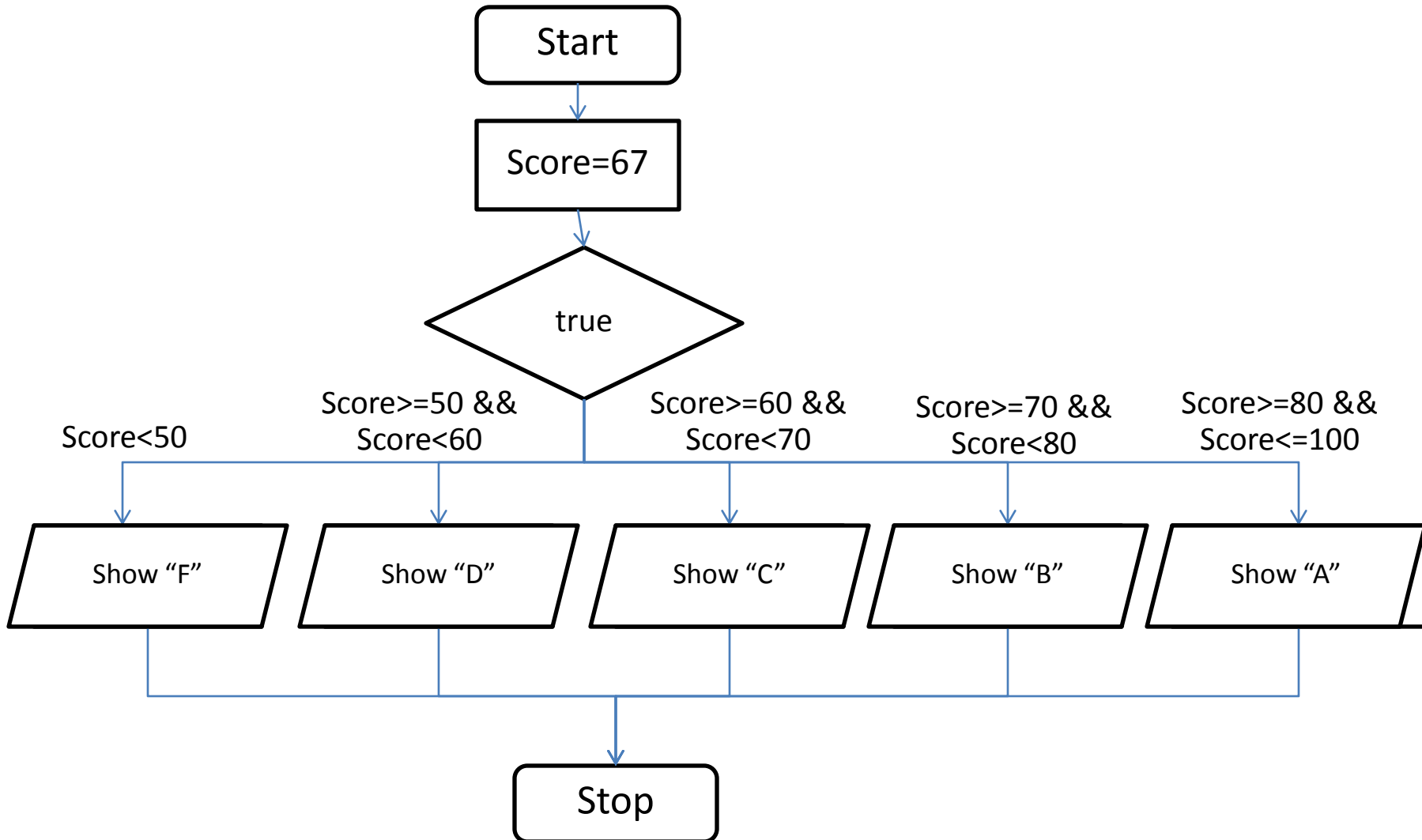
switch(true){
    case ($t<12):
        echo "ก่อนเที่ยง";
        break;
    case ($t>12):
        echo "หลังเที่ยง";
        break;
    default:
        echo "ไม่มีเวลา";
}

?>
```

switch Statement LAB#3



switch Statement LAB#4



PHP Loops

- In PHP, we have the following looping statements:
- **while** - loops through a block of code as long as the specified condition is true
- **do...while** - loops through a block of code once, and then repeats the loop as long as the specified condition is true
- **for** - loops through a block of code a specified number of times
- **foreach** - loops through a block of code for each element in an array

The PHP while Loop

```
while (condition is true) {  
    code to be executed;  
}
```

The PHP while Loop

Example

```
<?php
$x=1;

while($x<=5) {
    echo "The number is: $x <br>";
    $x++;
}
?>
```

The PHP do...while Loop

```
do {  
    code to be executed;  
} while (condition is true);
```

The PHP do...while Loop

Example

```
<?php
$x=1;

do {
    echo "The number is: $x <br>";
    $x++;
} while ($x<=5);
?>
```

The PHP for Loop

```
for (init counter; test counter; increment  
counter) {  
    code to be executed;  
}
```

The PHP for Loop

Example

```
<?php
for ($x=0; $x<=10; $x++) {
    echo "The number is: $x <br>";
}
?>
```


The PHP foreach Loop

```
foreach ($array as $value) {  
    code to be executed;  
}
```

The PHP foreach Loop

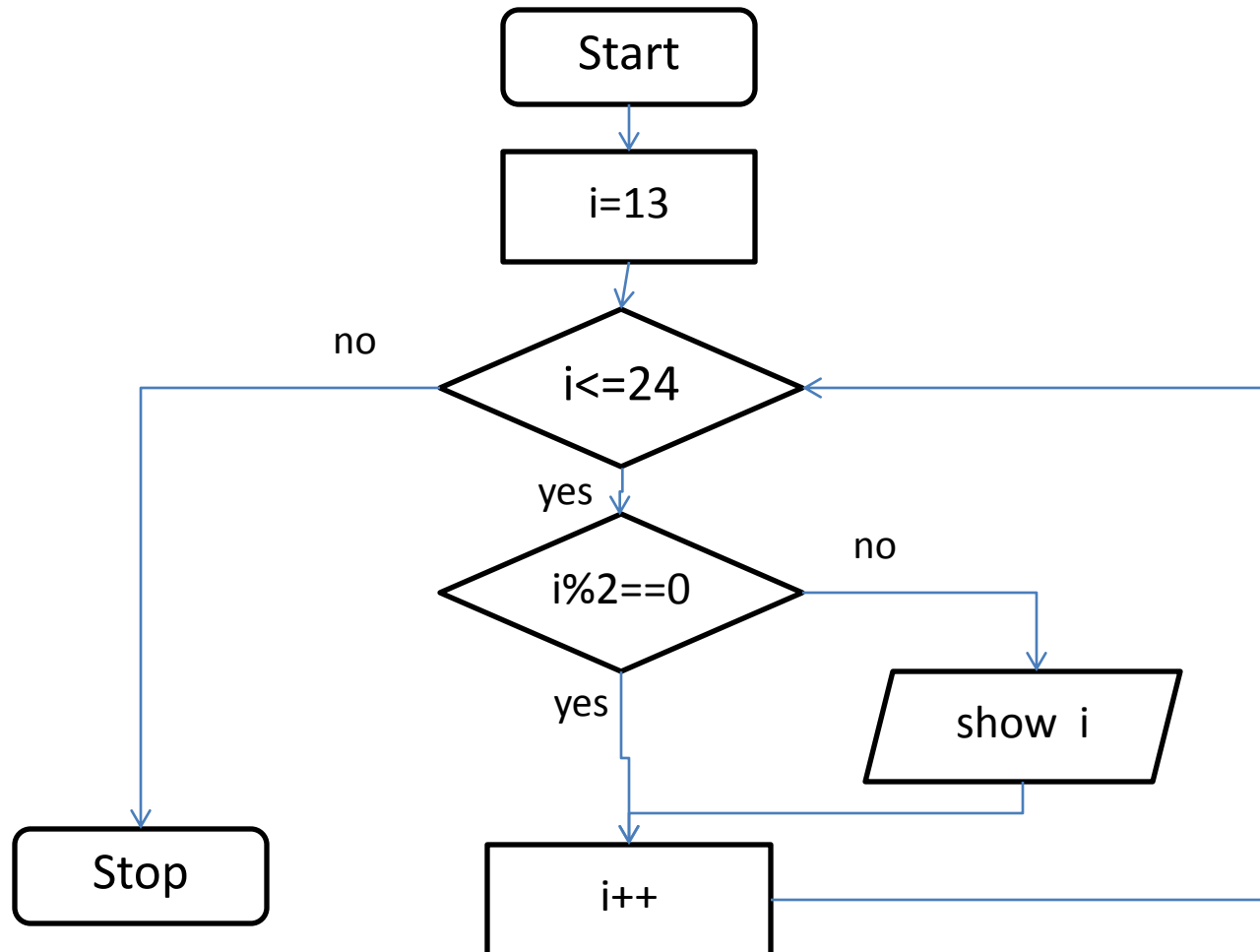
Example

```
<?php
$colors = array("red","green","blue","yellow");

foreach ($colors as $value) {
    echo "$value <br>";
}
?>
```

PHP Loop LAB#5

- แสดงตัวเลขคี่ ที่อยู่ระหว่างค่า 13 ถึง 24



PHP Functions

- PHP User Defined Functions
- Besides the built-in PHP functions, we can create our own functions.
- A function is a block of statements that can be used repeatedly in a program.
- A function will not execute immediately when a page loads.
- A function will be executed by a call to the function.

Create a User Defined Function in PHP

```
function functionName() {  
    code to be executed;  
}
```

PHP Function Arguments

Example

```
<?php
function writeMsg() {
    echo "Hello world!";
}

writeMsg(); // call the function
?>
```

PHP Default Argument Value

Example

```
<?php
function familyName($fname) {
    echo "$fname Refsnes.<br>";
}

familyName("Jani");
familyName("Hege");
familyName("Stale");
familyName("Kai Jim");
familyName("Borge");
?>
```

PHP Default Argument Values

Example

```
<?php
function familyName($fname,$year) {
    echo "$fname Refsnes. Born in $year <br>";
}

familyName("Hege","1975");
familyName("Stale","1978");
familyName("Kai Jim","1983");
?>
```


PHP Default Argument Value

Example

```
<?php
function setHeight($minheight=50) {
    echo "The height is : $minheight <br>";
}

setHeight(350);
setHeight(); // will use the default value of 50
setHeight(135);
setHeight(80);
?>
```

PHP Functions - Returning values

Example

```
<?php
function sum($x,$y) {
    $z=$x+$y;
    return $z;
}

echo "5 + 10 = " . sum(5,10) . "<br>";
echo "7 + 13 = " . sum(7,13) . "<br>";
echo "2 + 4 = " . sum(2,4);
?>
```

PHP Functions LAB#6

- ทำฟังก์ชัน ลบ คูณ หาร
 - Minus (x,y)
 - Multiply (x,y)
 - Divide (x,y)

PHP Arrays

- An array stores multiple values in one single variable

Create an Array in PHP

- In PHP, there are three types of arrays:
 - **Indexed arrays** - Arrays with numeric index
 - **Associative arrays** - Arrays with named keys
 - **Multidimensional arrays** - Arrays containing one or more arrays

Indexed arrays

Example

```
<?php
$cars=array("Volvo","BMW","Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>
```

Get The Length of an Array

- The **count()** Function

Example

```
<?php  
$cars=array("Volvo","BMW","Toyota");  
echo count($cars);  
?>
```

Loop Through an Indexed Array

Example

```
<?php
$cars=array("Volvo","BMW","Toyota");
$arlength=count($cars);

for($x=0;$x<$arlength;$x++) {
    echo $cars[$x];
    echo "<br>";
}
?>
```


PHP Associative Arrays

Example

```
<?php
$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");
echo "Peter is " . $age['Peter'] . " years old.";
?>
```

Loop Through an Associative Array

Example

```
<?php
$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");

foreach($age as $x=>$x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

PHP - Sort Functions For Arrays

- In this chapter, we will go through the following PHP array sort functions:
 - `sort()` - sort arrays in ascending order
 - `rsort()` - sort arrays in descending order
 - `asort()` - sort associative arrays in ascending order, according to the value
 - `ksort()` - sort associative arrays in ascending order, according to the key
 - `arsort()` - sort associative arrays in descending order, according to the value
 - `krsort()` - sort associative arrays in descending order, according to the key

Sort Array in Ascending Order - sort()

```
<?php
$cars=array("Volvo", "BMW", "Toyota");
sort($cars);

$clength=count($cars);
for($x=0;$x<$clength;$x++)
{
    echo $cars[$x];
    echo "<br>";
}
?>
```

BMW
Toyota
Volvo

Sort Array in Ascending Order - sort()

```
<?php
$numbers=array(4,6,2,22,11);
sort($numbers);
```

```
$arlength=count($numbers);
for($x=0;$x<$arlength;$x++)
{
    echo $numbers[$x];
    echo "<br>";
}
?>
```

2

4

6

11

22

Sort Array in Descending Order - rsort()

```
<?php
$cars=array("Volvo","BMW","Toyota");
rsort($cars);

$clength=count($cars);
for($x=0;$x<$clength;$x++)
{
    echo $cars[$x];
    echo "<br>";
}
?>
```

Volvo
Toyota
BMW

Sort Array in Descending Order - rsort()

```
<?php
$numbers=array(4,6,2,22,11);
rsort($numbers);

$arlength=count($numbers);
for($x=0;$x<$arlength;$x++)
{
    echo $numbers[$x];
    echo "<br>";
}
?>
```

22
11
6
4
2

Sort Array in Ascending Order(Value)- asort()

```
<?php
$age=array("Peter"=>"50","Ben"=>"47","Joe"=>"33");
asort($age);

foreach($age as $x=>$x_value)
{
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```


Sort Array in Ascending Order (Key) - ksort()

```
<?php
$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");
ksort($age);

foreach($age as $x=>$x_value)
{
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

Key=Ben, Value=37
Key=Joe, Value=43
Key=Peter, Value=35

Sort Array in Descending Order (Value) - arsort()

```
<?php
$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");
arsort($age);

foreach($age as $x=>$x_value)
{
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

Key=Joe, Value=43

Key=Ben, Value=37

Key=Peter, Value=35

Sort Array in Descending Order (Key) - krsort()

```
<?php
$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");
krsort($age);

foreach($age as $x=>$x_value)
{
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

Key=Peter, Value=35

Key=Joe, Value=43

Key=Ben, Value=37

PHP Global Variables - Superglobals

- The PHP superglobal variables are:
 - `$GLOBALS`
 - `$_SERVER`
 - `$_REQUEST`
 - `$_POST`
 - `$_GET`
 - `$_FILES`
 - `$_ENV`
 - `$_COOKIE`
 - `$_SESSION`

\$GLOBALS

Example

```
<?php
$x = 75;
$y = 25;

function addition() {
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}

addition();
echo $z;
?>
```

\$_SERVER

Example

```
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```

Collect Data after submitting an HTML Form

- `$_REQUEST`
- `$_POST`
- `$_GET`