1. **Decision-making structures cannot be nested. True or false with example?**
   * **Answer**: **False**  
     **Explanation**: Decision-making structures (such as if, else, and switch) **can** be nested. Nesting means placing one decision-making structure inside another.

**Example**: Nested if statements:

Explain

if ($age > 18) {

if ($hasID) {

echo "You can enter.";

} else {

echo "ID required.";

}

} else {

echo "You are too young to enter.";

}

1. **How do you perform more than one statement when a condition is true?**
   * **Answer**: To perform more than one statement when a condition is true, you can enclose the statements inside a block using curly braces {}. All the code inside the block will be executed if the condition evaluates to true.

**Example**:

Explain

if ($x > 10) {

$y = $x \* 2;

$z = $y + 10;

echo "The value of z is $z";

}

1. **Correct the syntax error line by line:**

Explain

<?php

1) if ($x > 25) // "If" should be lowercase

2) {

3) $Y = $x; // Variable $x should have a "$" sign

4) } else // Closing bracket was missing before "else"

5) $y = $z; // "%" symbol is invalid for variable declaration

6) ?>

**Corrected Code**:

Explain

<?php

if ($x > 25) {

$Y = $x;

} else {

$y = $z;

}

?>

1. **What is the function of the break keyword/statement and where can we use it?**
   * **Answer**: The break keyword is used to exit a loop or a switch statement prematurely, stopping further execution of the block.

**Example**: In a switch statement:

Explain

switch ($day) {

case 'Monday':

echo "Start of the week!";

break; // Exit the switch statement

case 'Friday':

echo "Almost weekend!";

break;

default:

echo "Regular day";

break;

}

**Example in a loop**:

Explain

while ($x < 10) {

if ($x == 5) {

break; // Exit the loop when $x equals 5

}

$x++;

}

1. **What is the function of the continue keyword/statement and where can we use it?**
   * **Answer**: The continue keyword is used within loops to skip the current iteration and jump to the next iteration of the loop. It does not exit the loop but skips the code below it for the current loop cycle.

**Example** in a for loop:

Explain

for ($i = 0; $i < 10; $i++) {

if ($i == 5) {

continue; // Skip the rest of the loop when $i equals 5

}

echo $i; // This will print all numbers except 5

}