Date	Sep-26-2024	Session No	3

Topic : Programming (C# and Dot Net) Theory + Practice Questions

Theory: Basics of C# Programming and Loops

Basics of C# Syntax

C# (pronounced "C-sharp") is a strongly typed, object-oriented language developed by Microsoft as part of the .NET framework.

- **Namespace Declaration**: Defines a scope to organize code and prevent name conflicts.
- **Class Definition**: Blueprint for creating objects.
- **Main Method**: The entry point of any C# application, defined as `static void Main(string[] args)` in a class.

Variables: Declare variables with specific data types like `int`, `string`, `float`, etc.

```
int number = 5;
string name = "John";
```

- **Data Types**: Common types include `int`, `string`, `char`, `bool`, `double`, `float`, etc.
- **Control Structures**: Includes conditional statements (`if`, `else`, `switch`) and loops.

Loops in C#

Loops are used to execute a block of code repeatedly. There are several types of loops in C#:

For Loop:

Executes a block of code a set number of times.

Syntax:

```
for (initialization; condition; iteration)
{
    // Code to execute
}
```

Example:

```
for (int i = 0; i < 5; i++)
{
    Console.WriteLine(i);
}
```

While Loop:

Repeats a block of code while a condition is true.

Syntax:

```
while (condition)
{
    // Code to execute
}
```

Example:

```
int i = 0;
while (i < 5)
{
    Console.WriteLine(i);
    i++;
}
```

Do-While Loop:

Similar to the `while` loop, but it checks the condition after the execution of the loop.

Syntax:

```
do
{
    // Code to execute
} while (condition);
```

Example:

```
int i = 0;
do
{
    Console.WriteLine(i);
    i++;
} while (i < 5);
```

Foreach Loop:

Used to iterate over the elements in a collection (arrays, lists).

Syntax:

```
foreach (var item in collection)
{
    // Code to execute
}
```

Example:

```
int[] numbers = { 1, 2, 3, 4, 5 };
foreach (int number in numbers)
{
    Console.WriteLine(number);
}
```

Practical Questions for Practice

For Loop Questions

Print Numbers from 1 to 10: Write a for loop to print the numbers from 1 to 10.

```
for (int i = 1; i <= 10; i++)
{
    Console.WriteLine(i);
}
```

Sum of First N Natural Numbers: Write a program that takes an integer n and calculates the sum of the first n natural numbers using a for loop.

```
int sum = 0;
for (int i = 1; i <= n; i++)
{
    sum += i;
}
Console.WriteLine("Sum = " + sum);
```

Print Multiplication Table: Write a program that prints the multiplication table of a given number.

```
int num = 5;
for (int i = 1; i <= 10; i++)
{
    Console.WriteLine("{0} x {1} = {2}", num, i, num * i);
}
```

While Loop Questions

Factorial of a Number: Write a program that calculates the factorial of a number using a while loop.

```
int number = 5;
int factorial = 1;
int i = 1;
```

```
while (i <= number)
{
    factorial *= i;
    i++;
}
Console.WriteLine("Factorial = " + factorial);
```

Guessing Game: Write a guessing game where the user needs to guess a number between 1 and 10. The game continues until the correct number is guessed.

```
int numberToGuess = 7;
int userGuess;
do
{
    Console.WriteLine("Guess the number:");
    userGuess = Convert.ToInt32(Console.ReadLine());
} while (userGuess != numberToGuess);
Console.WriteLine("Correct! You guessed the number.");
```

Do-While Loop Questions

Repeat User Input Until Correct: Write a program that repeatedly asks the user to enter the word "hello" until they enter it correctly.

```
string input;
do
{
    Console.WriteLine("Enter 'hello':");
    input = Console.ReadLine();
} while (input != "hello");
Console.WriteLine("You entered it correctly!");
```

Sum of Digits: Write a program to find the sum of digits of a number using a `do-while` loop.

```
int num = 123;
int sum = 0;
do
{
    sum += num % 10;
    num /= 10;
} while (num > 0);
Console.WriteLine("Sum of digits = " + sum);
```

Foreach Loop Questions

Print Array Elements: Given an array, print all its elements using a `foreach` loop.

```
int[] arr = { 1, 2, 3, 4, 5 };
foreach (int i in arr)
{
    Console.WriteLine(i);
}
```

Sum of Elements in Array: Write a program that calculates the sum of all elements in an array using the `foreach` loop.

```
int[] numbers = { 10, 20, 30, 40 };

int sum = 0;
foreach (int num in numbers)
{
    sum += num;
}
Console.WriteLine("Sum = " + sum);
```

Find Maximum Element in Array: Write a program to find the maximum element in an array using the `foreach` loop.

```
int[] arr = { 1, 5, 3, 9, 2 };
int max = arr[0];
foreach (int num in arr)
{
    if (num > max)
    {
        max = num;
    }
}
Console.WriteLine("Maximum = " + max);
```

Summary of C# Loops

- **For Loop**: Best for known iteration counts.
- **While Loop**: Used when the number of iterations is not known beforehand.
- **Do-While Loop**: Ensures that the loop is executed at least once.
- **Foreach Loop**: Efficient for iterating over collections like arrays or lists.

END