**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LOOPING STATEMENTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

What is meant by looping?

Looping in programming language is a way to execute a statement or a

set of statements multiple number of times depending on the result of condition

to be evaluated to execute statements. The result condition should be true to

execute statements within loops. There are 4 looping staements.They are 1.While 2.DoWhile 3.For 4.Foreach

**Define While ?**

While Loop:-

It repeats a statement or a group of statements while a given condition is true.

It tests the condition before executing the loop body.

\* While loop is used to execute a block of statements until the specified expression

return as a true.

**Syntax:-**

while keyword is used to create a while loop.

while (boolean\_expression) {

// Statements to Execute

}

Here,boolean\_expression returns true,then the statements inside of while loop will

be executed. After executing the statements, again the boolean\_expression will be

evaluated to execute the statements within the while loop.

the boolean\_expression is evaluated to false, then the while loop stops

execution of statements and the program comes out of the loop.

**Example:-**

namespace program1

{

class Program

{

static void Main(string[] args)

{

int i = 1;

while (i <= 4)

{

Console.WriteLine("i value: {0}", i);

i++;

}

Console.ReadLine();

}

}

**Nested While:**

Syntax:-

while(condition 1){

while(condition2) {

////statements(s)

}}

**Define dowhile ?**

It is similar to a while statement, except that it tests the condition at the end of the loop body

Syntax:-

do {

statement(s);

} while( condition );

//Here the loop execute at least one time

namespace Loops {

class Program {

static void Main(string[] args) {

// local variable definition

int a = 10;

// do loop execution

do {

Console.WriteLine("value of a: {0}", a);

a = a + 1;

}

while (a < 20);

Console.ReadLine();

}

}

}

**Nested dowhile:-**

do

{

statements(s)

do

{

statements(s)

} while (condition);

} while (condition2);

**Define For ?**

It executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.

Syntax:-

for (initialization; condition; iterator(inc / dec))

{

Statements

}

Example:-

namespace Loops {

class Program {

static void Main(string[] args) {

/\* for loop execution \*/

for (int a = 10; a < 20; a = a + 1) {

Console.WriteLine("value of a: {0}", a);

}

Console.ReadLine();

}

}

}

**For Loop with Multiple Variables:-**

we can declare and initialize multiple variables and iterator expressions by separating with comma (,) operator.

Syntax

static void Main(string[] args)

{

for (int i = 1, j = 0; i <= 4; i++, j++)

{

Console.WriteLine("i: {0}, j: {1}", i, j);

}

}

**For Loop without Initialization & Iterators:-**

namespace loops{

class Program

{

static void Main(string[] args) {

int i = 1;

for ( ; i <= 4; ) {

i++;

Console.WriteLine("i value: {0}", i);

}

Console.ReadLine();

}

}

}

**Infinite For Loop:-**

for ( ; ; ){

statements

}

**Nested For Loop:**

for (initialization; condition; iterator(inc / dec))

{

for (initialization; condition; iterator(inc / dec))

{

Statements

}

}

**Define Foreach?**

Foreach loop is useful to loop through each item in array or collection object to execute the block of statements repeatedly.

**Syntax**

foreach (Type var\_name in Collection\_Object) {

staements

}

**example**

class Program{

static void Main(string[] args) {

string[] names = new string[3] { "Suresh Dasari", "Rohini Alavala", "Trishika Dasari" };

foreach (string name in names) {

Console.WriteLine(name);

}

Console.WriteLine("Press Enter Key to Exit..");

Console.ReadLine();

}

}

}