

C#.NET Programming Lab Manual

1. to demonstrate nested if statement

Aim: To write a C# program using nested if statement.

Algorithm:

1. Start the program
2. Create a class name 'nestedif' and declare the variables.
3. Get the input from the user.
4. Check the condition using if and print the result.
5. Stop the process.

Coding:

```
using System;
public class nestedif {
public static void Main() {
int num1, num2, num3;
Console.WriteLine("Find the largest of three numbers:\n");
Console.WriteLine("Input the 1st number :");
num1 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Input the 2nd number :");
num2 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Input the 3rd number :");
num3 = Convert.ToInt32(Console.ReadLine());
if (num1 > num2) {
if (num1 > num3) {
Console.WriteLine("The 1st Number is the greatest among three. \n\n");}
Else {
Console.WriteLine("The 3rd Number is the greatest among three. \n\n");}}
else if (num2 > num3)
Console.WriteLine("The 2nd Number is the greatest among three \n\n");
else
Console.WriteLine("The 3rd Number is the greatest among three \n\n"); } }
```

Result: Thus the program is compiled and executed successfully and the results are obtained.

2.Program to demonstrate else if ladder

Aim: To write a C# program using else if ladder.

Algorithm:

1. Start the program
2. Create a class named 'elseifladder' and declare the variables.
3. Get the input from the user.
4. Check the condition using if and print the result.
5. Stop the process.

Coding:

```
using System;
public class elseifladder {
public static void Main(String[] args) {
int x, y, z=0,ch;
Console.WriteLine("Input the 1st number :");
x = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Input the 2nd number :");
y = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter your choice :");
ch = Convert.ToInt32(Console.ReadLine());
```

C#.NET Programming Lab Manual

```
if (ch == 1){
z = x + y;}
else if (ch == 2){
z = x - y;}
else if(ch == 3){
z = x * y; }
else if(ch == 4){
z = x/y ;}
else{
Console.WriteLine("\n Invalid choice! Please try again!");}
Console.WriteLine("Answer is {0}",z);}}
```

Result: Thus the program is compiled and executed successfully and the results are obtained.

3.Program to find sum of series using foreach statement

Aim: To write a C# program to find the sum of series using foreach statement.

Algorithm:

1. Start the program
2. Create a class named 'sumseries' and declare the array variables.
3. Using foreach loop to compute the sum of series
4. Print the result
5. Stop the process.

Coding:

```
using System;
using System.Linq;
public class sumseries {
public static void Main(String[] args) {
int[] number = new int[5];
number[0] = 542;
number[1] = 320;
number[2] = 138;
number[3] = 120;
number[4] = 238;
foreach(int turd in number) {
Console.WriteLine(turd); }
Console.WriteLine("Total= "+number.Sum()); }}
```

Result: Thus the program is compiled and executed successfully and the results are obtained.

4.Program to perform various arithmetic operations using switch statement

Aim: To write a C# program to perform various arithmetic operations using switch statement.

Algorithm:

1. Start the program
2. Create a class named 'arithmetic' and declare the variables
3. Perform the arithmetic operations using switch statement
4. Print the result
5. Stop the process.

Coding:

```
using System;
public class arithmetic {
public static void Main(String[] args) {
Console.Write("Enter No1: ");
int a = Convert.ToInt16(Console.ReadLine());
```

C#.NET Programming Lab Manual

```
Console.Write("Enter No2: ");
int b = Convert.ToInt16(Console.ReadLine());
Console.WriteLine("1.Addition");
Console.WriteLine("2.Subtraction");
Console.WriteLine("3.Divsion");
Console.WriteLine("4.Multiplication");
Console.WriteLine("5.Increment");
Console.WriteLine("6.Decrement");
int c = Convert.ToInt16(Console.ReadLine());
switch(c) {
case 1:
Console.WriteLine("Addition Of Two Numbers : "+(a+b));
break;
case 2:
Console.WriteLine("Subtraction Of Two Numbers : " + (a - b));
break;
case 3:
Console.WriteLine("Division Of Two Numbers : " + (a / b));
break;
case 4:
Console.WriteLine("Multiplcaion Of Two Numbers : " + (a * b));
break;
case 5:
Console.WriteLine("Increment Numbers : " + (++a) + " " + (++b));
break;
case 6:
Console.WriteLine("Decrement Numbers : " + (--a)+" "+(--b));
break;
default:
Console.WriteLine("Choose Only 1 To 6 ");
break; }
Console.ReadLine(); } }
```

Result: Thus the program is compiled and executed successfully and the results are obtained.

5.Program to print the reverse order of a given number using do....while

Aim: To write a C# program to print the reverse order of a given number.

Algorithm:

1. Start the program
2. Create a class named 'revorder' and declare the variables.
3. Get the input from the user.
4. Perform the calculations using do....while loop.
5. Print the result
6. Stop the process

Coding:

```
using System;
public class revorder {
public static void Main(String[] args) {
int n,a,r,s=0;
Console.Write("Input the number :");
n = Convert.ToInt32(Console.ReadLine());
```

C#.NET Programming Lab Manual

```
a=n;
do {
r=n%10;
s=s*10+r;
n=n/10;
}while(n>0);
Console.WriteLine("\n The Reverse Number of {0} is {1}",a,s);
}}
```

Result: Thus the program is compiled and executed successfully and the results are obtained.

6.Program to sort and to reverse the numbers using sort function

Aim: To write a C# program to sort and reverse the numbers using sort function.

Algorithm:

1. Start the program
2. Create a class named 'sortrev' and declare the array variables.
3. Using in-built array functions, sort and reverse the given array values.
4. Using foreach statement, print the values.
5. Stop the process.

Coding:

```
using System;
public class sortrev {
public static void Main() {
int[] arr = new int[] {1, 9, 6, 7, 5, 19};
Array.Sort(arr);
Array.Reverse(arr);
foreach(int value in arr) {
Console.Write(value + " ");}}}
```

Result: Thus the program is compiled and executed successfully and the results are obtained.

7.Program to demonstrate ArrayList Class

Aim: To write a C# program using ArrayList class.

Algorithm:

1. Start the program
2. Create a class named 'arraylist' and declare the array list.
3. Using the in-built functions of ArrayList, print the result
4. Stop the process.

Coding:

```
using System;
using System.Collections;
public class arraylist {
public static void Main(String[] args){
ArrayList myList = new ArrayList(15);
myList.Add(2);
myList.Add(4);
myList.Add(6);
myList.Add(8);
myList.Add(10);
myList.Add(12);
myList.Add(14);
myList.Add(16);
myList.Add(18);
```

C#.NET Programming Lab Manual

```
myList.Add(20);
Console.WriteLine("Count : " + myList.Count);
Console.WriteLine("Capacity : " + myList.Capacity);
Console.WriteLine(myList.IsFixedSize);
Console.WriteLine(myList.IsReadOnly);
myList.RemoveRange(0, 4);
Console.WriteLine("The ArrayList after Removing elements: ");
foreach(int i in myList) {
    Console.WriteLine(i); }
}}
```

Result: Thus the program is compiled and executed successfully and the results are obtained.

8.Program to find the vowels of a given string using string handling functions

Aim: To write a C# program to find the vowels of a given string.

Algorithm:

1. Start the program
2. Create a class named 'vowcount' and declare the variables.
3. Check the condition using if and print the result
4. Stop the process.

Coding:

```
using System;
public class vowcount {
    public static void Main() {
        string myStr;
        int i, len, vowel_count, cons_count;
        myStr = "hajiram beevi";
        vowel_count = 0;
        cons_count = 0;
        len = myStr.Length;
        for(i=0; i<len; i++) {
            if(myStr[i] == 'a' || myStr[i] == 'e' || myStr[i] == 'i' || myStr[i] == 'o' || myStr[i] == 'u' ||
            myStr[i] == 'A' || myStr[i] == 'E' || myStr[i] == 'I' || myStr[i] == 'O' || myStr[i] == 'U') {
                vowel_count++;
            } else {
                cons_count++; } }
        Console.WriteLine("\nVowels in the string: {0}\n", vowel_count); } }
```

Result: Thus the program is compiled and executed successfully and the results are obtained.

9.Program to demonstrate multilevel inheritance

Aim: To write a C# program using multilevel inheritance.

Algorithm:

1. Start the program
2. Create a class named 'MulInherit' and create an object for the derived class 'derive2'
3. Create a base class named 'bas1' and declare the variable and define the method 'getdata()'.
4. Create a derived class named 'derive1' and declare the variable and define the method 'readdata()'.
5. Create a derived class named 'derive2' and declare the variable and define the methods 'indata() and product()'.
6. Call the methods using objects.
7. Display the result.

C#.NET Programming Lab Manual

8. Stop the process.

Coding:

```
using System;
public class MulInherit {
public static void Main(String[] args) {
derive2 a=new derive2();
a.getdata();
a.readdata();
a.indata();
a.product(); } }
class bas1 {
public int x;
public void getdata() {
Console.Write("Enter value of x");
x = Convert.ToInt32(Console.ReadLine()); } }
class derive1 : bas1 {
public int y;
public void readdata() {
Console.Write("Enter value of y");
y = Convert.ToInt32(Console.ReadLine()); }}
class derive2 : derive1 {
public int z;
public void indata() {
Console.Write("Enter value of z");
z = Convert.ToInt32(Console.ReadLine()); }
public void product() {
int u=x*y*z;
Console.Write("{0}",u); } }
```

Result: Thus the program is compiled and executed successfully and the results are obtained.

10.Program to create and implement a delegate

Aim: To write a C# program using a delegate.

Algorithm:

1. Start the program
2. Create a class named 'delgate' and declare the delegate functions addnum() and subnum().
3. Create an object for two delegate functions addnum and subnum.
4. Pass the values to delegate functions
5. Perform the calculations
6. Print the result
7. Stop the process

Coding:

```
using System;
public class delgate {
public delegate void addnum(int a, int b);
public delegate void subnum(int a, int b);
public void sum(int a, int b) {
Console.WriteLine("(100 + 40) = {0}", a + b); }
public void subtract(int a, int b) {
Console.WriteLine("(100 - 60) = {0}", a - b); }
```

C#.NET Programming Lab Manual

```
public static void Main(String []args) {
    delgate obj = new delgate();
    addnum del_obj1 = new addnum(obj.sum);
    subnum del_obj2 = new subnum(obj.subtract);
    del_obj1(100, 40);
    del_obj2(100, 60); } }
```

Result: Thus the program is compiled and executed successfully and the results are obtained.

11.Program using try and catch for exception handling

Aim: To write a C# program using try and catch for exception handling.

Algorithm:

1. Start the program
2. Create a class named 'DivNumbers' and declare the variables.
3. Create a try, catch, finally blocks functions
4. Display the result
5. Stop the process.

Coding:

```
using System;
public class DivNumbers {
    int result;
    DivNumbers() {
        result = 0;}
    public void division(int num1, int num2) {
        try {
            result = num1 / num2;
        } catch (DivideByZeroException e) {
            Console.WriteLine("Exception caught: {0}", e);
        } finally {
            Console.WriteLine("Result: {0}", result);} }
    public static void Main(string[] args) {
        DivNumbers d = new DivNumbers();
        d.division(25, 0);} }
```

Result: Thus the program is compiled and executed successfully and the results are obtained.

12.Program to create and implement an event handler

Aim: To write a C# program to create and implement an event handler.

Algorithm:

1. Start the program
2. Create a class named 'EventClass' and declare a delegate function
3. Create a function and check the status.
4. Create a class named 'EventTest' and create object for both classes.
5. Call the methods using that objects.
6. Display the result.
7. Stop the program.

Coding:

```
using System;
public delegate void Edelegate(string str);
public class EventClass{
    public event Edelegate Status;
    public void TriggerEvent(){
        if(Status!=null)
```

C#.NET Programming Lab Manual

```
Status("Event Triggered");} }
public class EventTest{
public static void Main(String[] args){
EventClass ec=new EventClass();
EventTest et=new EventTest();
ec.Status+=new Edelegate(et.EventCatch);
ec.TriggerEvent();}
public void EventCatch(string str){
Console.WriteLine(str);} }
```

Result: Thus the program is compiled and executed successfully and the results are obtained.

13.Program to demonstrate the multithreading

Aim: To write a C# program using multithreading

Algorithm:

1. Start the program
2. Create a class named 'multthread' and declare the variables.
3. Define two methods 'method1' and 'method2' and invoke the thread.sleep() method
4. Print the result
5. Stop the process

Coding:

```
using System;
using System.Threading;
public class multthread {
public static void method1() {
for (int I = 0; I <= 10; I++) {
Console.WriteLine("Method1 is : {0}", I);
if (I == 5) {
Thread.Sleep(6000); } } }
public static void method2() {
for (int J = 0; J <= 10; J++) {
Console.WriteLine("Method2 is : {0}", J); } }
public static void Main(String[] args) {
Thread thr1 = new Thread(method1);
Thread thr2 = new Thread(method2);
thr1.Start();
thr2.Start(); } }
```

Result: Thus the program is compiled and executed successfully and the results are obtained.

14. Develop a windows application for simple calculator

Aim: To build a calculator widget in windows application using C#.

Algorithm:

1. Create a windows application using C#.
2. Design the form with buttons and textbox like a calculator.
3. Name the buttons using property window.
4. Write the code for each button.
5. Build the application.
6. Display the result.
7. Stop the program

Coding:

```
using System;
using System.Collections.Generic;
```


C#.NET Programming Lab Manual

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
namespace WindowsApplication1
{
    public partial class frmcalc : Form
    {
        private string s1, s2, oper;
        private bool isoper;
        public frmcalc(){
            InitializeComponent();
            s1 = s2 = oper = " ";
            isoper = true;}
        private void button17_Click(object sender, EventArgs e){
            s1 = s2 = oper = " ";
            isoper = true;
            txtResult.Text = s1;
            this.dot.Enabled = false;}
        private void zero_Click(object sender, EventArgs e){
            s1 += "0";
            txtResult.Text = s1;}
        private void one_Click(object sender, EventArgs e){
            s1 += "1";
            txtResult.Text = s1;}
        private void two_Click(object sender, EventArgs e){
            s1 += "2";
            txtResult.Text = s1;}
        private void three_Click(object sender, EventArgs e){
            s1 += "3";
            txtResult.Text = s1;}
        private void four_Click(object sender, EventArgs e){
            s1 += "4";
            txtResult.Text = s1;}
        private void five_Click(object sender, EventArgs e){
            s1 += "5";
            txtResult.Text = s1;}
        private void six_Click(object sender, EventArgs e){
            s1 += "6";
            txtResult.Text = s1;}
        private void seven_Click(object sender, EventArgs e){
            s1 += "7";
            txtResult.Text = s1;}
        private void eight_Click(object sender, EventArgs e){
            s1 += "8";
            txtResult.Text = s1;}
        private void div_Click(object sender, EventArgs e){
            oper = "/";
```

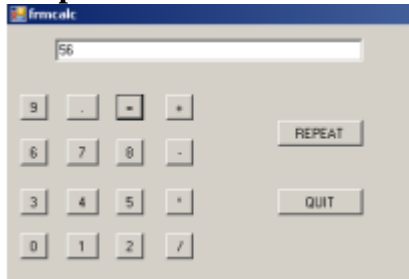
C#.NET Programming Lab Manual

```
s2 = s1;
s1 = null;
txtResult.Text = oper;
isoper = true;
this.dot.Enabled = true;}
private void mul_Click(object sender, EventArgs e){
oper = "*";
s2 = s1;
s1 = null;
txtResult.Text = oper;
isoper = true;
this.dot.Enabled = true;}
private void sub_Click(object sender, EventArgs e){
oper = "-";
s2 = s1;
s1 = null;
txtResult.Text = oper;
isoper = true;
this.dot.Enabled = true;}
private void nine_Click(object sender, EventArgs e){
s1 += "9";
txtResult.Text = s1;}
private void add_Click(object sender, EventArgs e){
oper = "+";
s2 = s1;
s1 = null;
txtResult.Text = oper;
isoper = true;
this.dot.Enabled = true;}
private void dot_Click(object sender, EventArgs e){
if (isoper){
s1 += ".";
this.dot.Enabled = false;
isoper = false;
txtResult.Text = s1; }}
private void equal_Click(object sender, EventArgs e){
float n1, n2, res = 0.0f;
n1 = float.Parse(s2);
n2 = float.Parse(s1);
switch (oper) {
case "+":
res = n1 + n2;
break;
case "-":
res = n1 - n2;
break;
case "*":
res = n1 * n2;
break;
```

C#.NET Programming Lab Manual

```
case "/":  
res = n1 / n2;  
break;}  
this.txtResult.Text = res.ToString();  
isoper = true;}  
private void QUIT_Click(object sender, EventArgs e){  
MessageBox.Show("See You Again", "Thank You");  
this.Close();}}}
```

Output:



Result: Thus the program is compiled and executed successfully and the results are obtained.