

Lab Manual

Nandinee Bhatt

Contents

| | |
|-------------------|---|
| Practical-1 | 1 |
| Practical-2 | 1 |

Practical-1

AIM-

-Introduction to c#:

Variables:

 Initialization

 Scope

 Constant

-Predefined Data Types

 Value Types

 Reference TTypes

-Flow Control

 Conditional Statements(if, switch)

 Loop(for, while, dowhile, foreach)

 Jump(goto, break, continue, return)

-Eumerations

-Passing Arguments

```
using System;
namespace P1
{
    class MyFirstClass
    {
        public static void Main()
        {
            Console.WriteLine("HiAll");
            Console.ReadKey();
            return;
        }
    }
}
```

```

2.constant variable
using System;
namespace Cant
{
    public class Cant
    {
        public static void Main()
        {
            int a;
            a = 99;
            Console.WriteLine("Value is: {0}",a);

            Console.ReadKey();
        }
    }
}

3.scope of variable
using System;
namespace P1
{
    class Scope1
    {
        public static void Main()
        {
            for(int i=0;i<5;i++)
            {
                Console.WriteLine(i);
            }

            //i goes out of Scope here

            for(int i=4;i>=0;i--)
            {
                Console.WriteLine(i);
            }
        }
    }
}

4.scope of variable
using System;
namespace P1
{
    class Scope2
    {
        public static void Main()
        {
            int j;
            for(int i=0;i<15;i++)
        }
}

```

```

        {
            int j;
            Console.WriteLine(i);
        }
    }
}
5.
using System;
namespace P1
{
    public class Scope{
        static int j = 430;
        public static void Main()
        {
            int j =900;
            Console.WriteLine(Scope.j);
        }
    }
6.consatnt variable
using System;
namespace P1
{
    public class Const
    {
        public static void Main()
        {
            const double bonusPercent = 0.51;
            int sal = 3000;
            int bonus = (int)(sal * bonusPercent);
            Console.WriteLine(bonus);
        }
    }
}
7.
using System;
namespace P1
{
    public class Vector
    {
        public int value;
    }
    public class DataTypes
    {
        public static void Main()
        {
            int i;
            int j;

```

```

        i = 77;
        j = i;

        Console.WriteLine("i is {0} and j is {1}", i, j);
        j = 20;
        Console.WriteLine("i is {0} and j is {1}", i, j);

        Vector x,y;
        x = new Vector();
        x.value = 33;
        y = x;
        Console.WriteLine("x is {0} and y is {1}", x.value, y.value);
        y.value = 24;
        Console.WriteLine("x is {0} and y is {1}", x.value, y.value);

    }
}
}

8.integer signed or unsigned variables
using System;
namespace P1
{
    class IntType
    {
        public static void Main()
        {
            //Signed Variables
            sbyte sb = 33;
            short s = 33 ;
            int i = 33;
            long l = 33L;

            //Unsigned Variables
            byte b = 33;
            ushort us = 33;
            uint ui = 33U;
            ulong ul = 33UL;
            us = (ushort)ul;

            Console.WriteLine("{0} {1} {2} {3} {4} {5} {6} {7}",
sb,s,i,l,b,us,ui,ul);

        }
    }
}

9.floating variables
using System;
namespace P1

```

```
{  
    public class Floatting  
    {  
        public static void Main()  
        {  
            float f = 0.123456789F;  
            double d = 0.112233445566778899;  
            decimal dec = 11223344.111222334445556667778889999M;  
            f = (float)d;  
            Console.WriteLine("f is {0} and d is {1} and dec is {2}", f, d, dec);  
        }  
    }  
}  
  
10.boolean  
using System;  
namespace P1  
{  
    public class Boolean  
    {  
        public static void Main()  
        {  
            bool status = true;  
            Console.WriteLine(status);  
        }  
    }  
}  
11.charcter  
using System;  
  
namespace P1  
{  
    public class Char  
    {  
        public static void Main()  
        {  
            char c = 'a';  
            Console.WriteLine(\a);  
        }  
    }  
}
```

Practical-2

AIM:GTU Programs

1)Write console based program in code behind language VB or C# to print following pattern.

```
@ @ @ @ @  
@ @ @ @  
@ @ @  
@ @  
@
```

```
using System;  
namespace Pattern  
{  
    class PatternExample  
    {  
        public static void Main()  
        {  
            int i,j=5;  
            for (; j > 0; j--)  
            {  
                for (i = j; i > 0; i--)  
                    Console.Write("@ ");  
                Console.WriteLine();  
            }  
        }  
    }  
}
```

2)Write console based program in code behind language VB or C# to print following pattern.

1
1 2
1 2 3
1 2 3 4

```
using System;
namespace Pattern
{
    class patternExample
    {
        public static void Main()
        {
            int i, j;
            for (j = 1; j < 5; j++)
            {
                for (i = 1; i <= j; i++)
                    Console.Write(i + " ");
                Console.WriteLine();
            }
        }
    }
}
```

3. Write C# code to prompt a user to input his/her name and country name and then the output will be shown as an example below:

Hello Ram from country India

```
using System;
public class userdata
{
    public static void Main()
    {
        string name, country;
        Console.Write("Enter Your Name: ");
        name = Console.ReadLine();
        Console.Write("Enter Your Country: ");
        country = Console.ReadLine();
        Console.WriteLine("Hello " + name + " from country " + country);
    }
}
```

4. What is inheritance? Create C# console application to define Car class and derive Maruti and Mahindra from it to demonstrate inheritance.

```
using System;
```

```
public class Car
{
    protected string name;
    public Car(string name)
    {
        this.name = name;
    }
    public Car()
    {
    }
    public virtual string Name
    {
        get{return name;}
        set
        {
            if(value.Length>3)
                name = value;
            else
                name="Unknown";
        }
    }
}

public class Maruti : Car
{
    public Maruti(string name) : base(name)
    {
    }
    public override string Name
    {
        get{return name;}
        set
        {
            if(value.Length>3)
                name = value + " -Maruti";
            else
                name="Unknown";
        }
    }
    public bool haveAGS;
}

public class Mahindra : Car
{
```

```
public Mahindra(string name) : base(name)
{
}
public Mahindra(){}
public override string Name
{
    get{return name;}
    set
    {
        if(value.Length>3)
            name = value + " -Mahindra";
        else
            name="Unknown";
    }
}
public class Program
{
    public static void Main()
    {
        Maruti car1 = new Maruti("Swift");
        car1.haveAGS = true; car1.Name = "Swift";
        Console.WriteLine("Details Car 1: {0} and
{1}",car1.Name,car1.haveAGS==true?"Have AGS":"not Have AGS");
        Mahindra car2 = new Mahindra();
        car2.Name = "XUV500";
        Console.WriteLine("Car 2: {0}",car2.Name);
    }
}
```