

Day_10 Assignment



SYS

C# OOP's concepts about inheritance and polymorphism.

Nalli Prudhvi.

@Nations Benefits

04-Feb-2022

Q. Example for single inheritance.

CODE

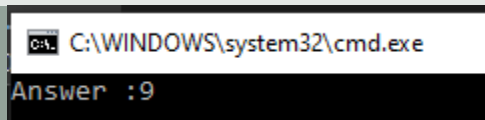
```
/// <summary>
/// RETURNS MATH VALUES
/// </summary>
class Math
{
    public int Add(int a, int b)
    {
        return a+ b;
    }
    public int Sub(int a , int b)
    {
        return a-b;
    }
}
/// <summary>
/// THIS RETURN THE MATHEMATICAL OPERATION OF ADV_MATH CLASS AND MATH VALUE.
/// </summary>
class Aplied_math : Math
{
    public int mul(int a, int b)
    {
        return a * b;
    }

    public int Div(int a, int b)
    {
        return a/b;
    }
}

internal class Program
{
    static void Main(string[] args)
    {
        /*****
        * AUTHOR :PRUDHVI
        * PURPOSE:EXAMPLE FOR SINGLE INHERITANCE
        * *****/
        var value = new Aplied_math();
        Console.WriteLine($"Answer :{ value.Div(45, 5)}");

        Console.ReadLine();
    }
}
```

OUTPUT



C:\WINDOWS\system32\cmd.exe
Answer :9

Q. Example for Multi inheritance

CODE

```
/// <summary>
/// RETURNS MATH VALUES
/// </summary>
class Math
{
    public int Add(int a, int b)
    {
        return a+ b;
    }
    public int Sub(int a , int b)
    {
        return a-b;
    }
}
/// <summary>
/// THIS RETURN THE MATHEMATICAL OPERATION OF ADV_MATH CLASS AND MATH VALUE.
/// </summary>
class Aplied_math : Math
{
    public int mul(int a, int b)
    {
        return a * b;
    }
    /// <summary>
    /// RETURNS a POWER b VLUE
    /// </summary>
    /// <param name="a">BASE</param>
    /// <param name="b">POWER</param>
    /// <returns></returns>

    public int Div(int a, int b)
    {
        return a/b;
    }
}
class Surds : Aplied_math
{
    public int Pwr(int a, int b ,int c=1)
    {
        for (int i = 0; i < b; i++)
        {
            c *= a;
        }
        return c;
    }
}

internal class Program
{
    static void Main(string[] args)
    {
        /*****
        * AUTHOR :PRUDHVI
        * PURPOSE:EXAMPLE FOR MULTI INHERITANCE
        * *****/
        var value = new Surds();
        Console.WriteLine("Enter your a value:");
```

```

    int a = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine();
    Console.Write("Enter your b value:");
    int b = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine();
    Console.WriteLine($"Answer a power b :{ value.Pwr(a, b)}");
    Console.WriteLine($"Answer a & b sum :{ value.Add(a, b)}");

    Console.ReadLine();
}
}

```

OUTPUT

```

C:\WINDOWS\system32\cmd.exe
Enter your a value:35
Enter your b value:6
Answer a power b :1838265625
Answer a & b sum :41

```

Q. Explain three types of inheritance in pictorial representation.

A.

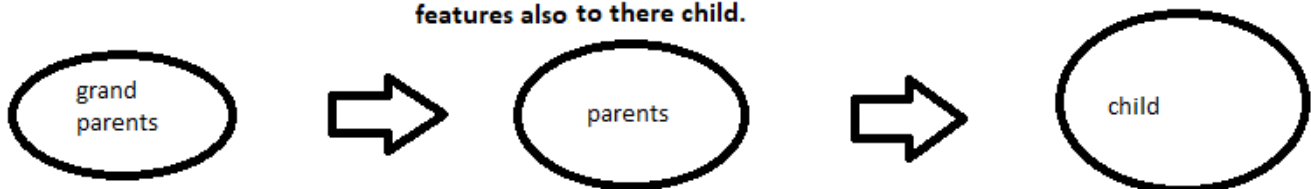
SINGLE INHERITANCE

Here parents are traiting there features with the child



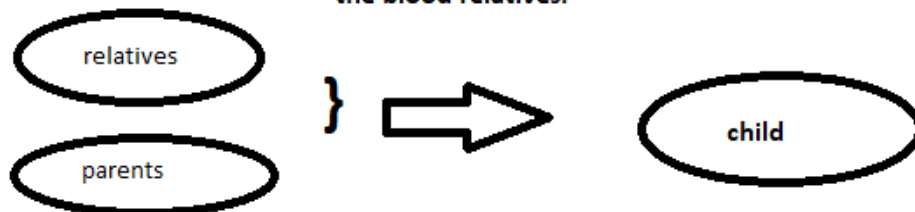
MULTI INHERUTANCE

Here grand parents traits features to children and they trait there feautures as well as there parents features also to there child.



MULTIPLE INHERITANC

Here the features are traited from parents and the blood relatives.



Q. Explain why In C# multiple inheritance is not supported.

- A. In C# compilers have a little ambiguity with multiple inheritance. Because of the diamond problem to be precise let see an example,. If two classes B and C inherit from A, and class D inherits from both B and C. If a method in D calls a method defined in A (and does not override it), and B and C have overridden that method differently, then via which class does it inherit: B, or C? So., multiple inheritance is not possible in C#. that is called Diamond Problem. But., in C# we can solve the Diamond problem with the help of interfaces in some case

Q. What is polymorphism?

- A. Polymorphism is often referred to as the third pillar of object-oriented programming, after encapsulation and inheritance.

At run time, objects of a derived class may be treated as objects of a base class in places such as method parameters and collections or arrays. When this polymorphism occurs, the object's declared type is no longer identical to its run-time type.

Q. Write a Sample Code for over loading.

CODE

```
/// <summary>
/// add over loaded
/// </summary>
class Overload
{
    public int Add(int a, int b )
    {
        return (a + b );
    }
    public int Add(int a, int b , int c)
    {
        return (a+b+c);
    }
    public float Add(float a, float b, float c )
    {
        c = a + b;
        return c;
    }
}
internal class Program
{
    static void Main(string[] args)
    {
        /*****
        * AUTHOR :PRUDHVI
        * PURPOSE:EXAMPLE FOR POLYMORPHISM overloading
        * *****/

        var ob1 = new Overload();

        Console.WriteLine($"answer :{ob1.Add(5,3)}");
    }
}
```

```
}
```

OUTPUT

```
CA. C:\WINDOWS\system32\cmd.exe
answer :8
Press any key to continue . . .
```

Q. write a example for overriding

CODE

```
/// <summary>
/// this class example for polymorphism
/// </summary>
class overriding
{
    public void Gm()
    {
        Console.WriteLine("Good Morning");
    }
    public void Hi()
    {
        Console.WriteLine("Hi");
    }
    public void Ge()
    {
        Console.WriteLine("good evening");
    }
    public void Bye()
    {
        Console.WriteLine("BYE");
    }
}
class Overriding_2 : overriding
{
    public new void Bye()
    {
        Console.WriteLine("ciao");
    }
}
internal class Program
{
    static void Main(string[] args)
    {
        /*****
        * AUTHOR :PRUDHVI
        * PURPOSE:EXAMPLE FOR POLYMORPHISM overriding with new key word
        * *****/

        var obj1 = new Overriding_2();
        obj1.Bye();
    }
}
```

OUTPUT

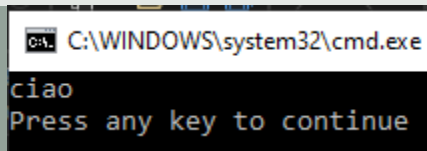
```
CA. C:\WINDOWS\system32\cmd.exe
ciao
Press any key to continue
```


CODE

```
// <summary>
/// this class example for polymorphism
/// </summary>
class overriding
{
    public void Gm()
    {
        Console.WriteLine("Good Morning");
    }
    public void Hi()
    {
        Console.WriteLine("Hi");
    }
    public void Ge()
    {
        Console.WriteLine("good evening");
    }
    public virtual void Bye()
    {
        Console.WriteLine("BYE");
    }
}
class Overriding_2 : overriding
{
    public override void Bye()
    {
        Console.WriteLine("ciao");
    }
}
internal class Program
{
    static void Main(string[] args)
    {
        /*****
        * AUTHOR :PRUDHVI
        * PURPOSE:EXAMPLE FOR POLYMORPHISM overriding with override and
        * Virtual keywords
        * *****/

        var obj1 = new Overriding_2();
        obj1.Bye();
    }
}
```

OUTPUT



C:\WINDOWS\system32\cmd.exe
ciao
Press any key to continue

-Thank you