Abstract Class : It is a class which contains some methods which are defined , some methods which are not defined.

Abstract class figure

{

Protected int area;

Abstract public void GetDeimensions();

Abtarct public void CalculateArea();

Public void DisplayArea()

{ Console.Write(area);

}

Class square : figure

{

Int side;

override public void GetDeimensions(){

Console.Write(“Enter Value of side”);

Side = Convert.ToByte(Console.ReadLine());

}

override public void CalculateArea(){ area = side \* side;}

Interface : It is syntactical contract within which we can only declare methods ,

we can not define them

2. We cannot declare variable

3. There is no access specifier

Differences in Abstract classes & Interface

1.Abstract Class : It’s a class in which some of the methods are defined and some of the methods are not defined

1.Interface : None of the methods can be defined

2. We cannot declare variables in Interface, but we can declare variables in Abstract Class

3. We do not add any access specifier with the methods in Interface

4. Interfaces are implemented , where as Abstract classes are inherited

5. Syntax is same

interface figure

{

void GetDimensions();

void CalclateArea();

void DisplayArea();

}

class rec1 : figure

{

public void CalclateArea()

{

}

public void DisplayArea()

{

}

public void GetDimensions()

{

}

}

class rec : figure

{

int l, b, area;

public void CalclateArea()

{

area = l \* b;

}

public void DisplayArea()

{

Console.WriteLine("Area is " + area);

}

public void GetDimensions()

{

Console.WriteLine("Enter Length");

l = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Breadth");

b = Convert.ToByte(Console.ReadLine());

}

}