**ASSIGNEMENT:--8(Abstract class)**

1.

using System;

namespace ConsoleApp3

{

abstract class Marks

{

public abstract float getPercentage();

}

class A : Marks

{

private int mark1, mark2, mark3;

public override float getPercentage()

{

float perc = ((mark1 + mark2 + mark3) / (float)300) \* 100;

return perc;

}

public A(int m1,int m2,int m3)

{

mark1 = m1;

mark2 = m2;

mark3 = m3;

}

}

class B : Marks

{

private int mark1, mark2, mark3;

public override float getPercentage()

{

float perc = ((mark1 + mark2 + mark3) / (float)300) \* 100;

return perc;

}

public B(int m1, int m2, int m3)

{

mark1 = m1;

mark2 = m2;

mark3 = m3;

}

}

class Program

{

static void Main(string[] args)

{

A ob1 = new A(70,80,90);

Console.WriteLine(ob1.getPercentage());

B ob2 = new B(90, 80, 90);

Console.WriteLine("Percentage is " +(ob2.getPercentage()));

}

}

}

3.

using System;

namespace ConsoleApp3

{

public abstract class Animals

{

public abstract void cats();

public abstract void dogs();

}

class Cats : Animals

{

public override void cats() //Inherited from abstract class

{

Console.WriteLine("Cats Meow");

}

public override void dogs()

{

}

}

class Dogs : Animals

{

public override void cats()

{

}

override public void dogs()

{

Console.WriteLine("Dogs bark");

}

}

class Program

{

static void Main(string[] args)

{

Cats c = new Cats();

c.cats();

Dogs d = new Dogs();

d.dogs();

}

}

}

2.