**AIM: Design Applications using Inheritance and Abstract Classes**

1. **Write a program to implement multilevel inheritance from the following figure. Accept and display data for one student.**

**SOURCE CODE:**

using System;

namespace P3MultiLevelInheritance

{

public class Student

{

int rollno;

string name;

public void getStudentData()

{

Console.WriteLine("Enter Roll No:");

rollno = Int32.Parse(Console.ReadLine());

Console.WriteLine("Enter Name:");

name = Console.ReadLine();

}

public void displayStudentData()

{

Console.WriteLine("Roll No: "+ rollno);

Console.WriteLine("Name: "+ name);

}

}

public class Test : Student

{

public int marks1, marks2;

public void getMarks()

{

getStudentData();

Console.WriteLine("Enter Marks 1:");

marks1 = Int32.Parse(Console.ReadLine());

Console.WriteLine("Enter Marks 2:");

marks2 = Int32.Parse(Console.ReadLine());

}

public void setMarks()

{

displayStudentData();

Console.WriteLine("Marks1: "+ marks1);

Console.WriteLine("Marks2: "+ marks2);

}

}

public class Result : Test

{

int calc;

public void getCalc()

{

getMarks();

calc = (marks1 + marks2)/2;

setMarks();

Console.WriteLine("Total: "+ calc);

}

}

class Program

{

static void Main(string[] args)

{

Result r = new Result();

r.getCalc();

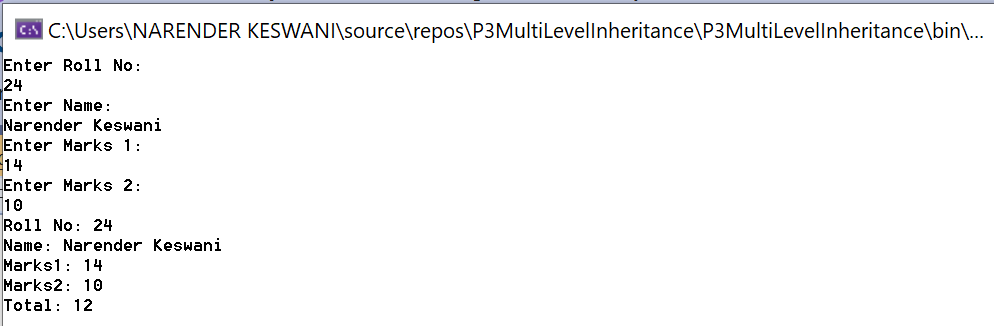
Console.ReadLine();

}

}

}

**OUTPUT:**



1. **Create a Super Class Student and two subclasses of it, Graduate and UnderGraduate. The members of the Student are name, id, grade, age and address and one method : boolean method IsPassed which takes in the parameter integer grade(0-100) and return true.The two subclasses override the method, for UG its 70% for passing and for G its 80% as passing grade.**

**SOURCE CODE:**

**using System;**

namespace P3Overriding

{

public class Student

{

public int id, grade, age;

public string name, address;

public void getData()

{

Console.WriteLine("Enter ID:");

id = Int32.Parse(Console.ReadLine());

Console.WriteLine("Enter Name:");

name = Console.ReadLine();

Console.WriteLine("Enter age:");

age = Int32.Parse(Console.ReadLine());

Console.WriteLine("Enter address:");

address = Console.ReadLine();

Console.WriteLine("Enter Grade:");

grade = Int32.Parse(Console.ReadLine());

}

public void displayData()

{

Console.WriteLine("ID: " + id);

Console.WriteLine("Name: "+ name);

Console.WriteLine("Age: "+ age);

Console.WriteLine("Address: "+ address);

Console.WriteLine("Grade: " + grade);

}

public virtual Boolean IsPassed()

{

return true;

}

}

public class Graduate : Student

{

public override bool IsPassed()

{

if (grade >= 80 && grade <= 100)

{

return true;

}

else

{

return false;

}

}

}

public class Undergraduate : Student

{

public override bool IsPassed()

{

if (grade >= 70 && grade <= 100)

{

return true;

}

else

{

return false;

}

}

}

class Program

{

static void Main(string[] args)

{

Graduate g = new Graduate();

g.getData();

g.displayData();

g.IsPassed();

if (g.IsPassed() == true)

{

Console.WriteLine("Passed in Graduation.");

}

else

{

Console.WriteLine("Failed in Graduation.");

}

Undergraduate ug = new Undergraduate();

ug.getData();

ug.displayData();

ug.IsPassed();

if (ug.IsPassed() == true)

{

Console.WriteLine("Passed in UnderGraduation.");

}

else

{

Console.WriteLine("Failed in UnderGraduation.");

}

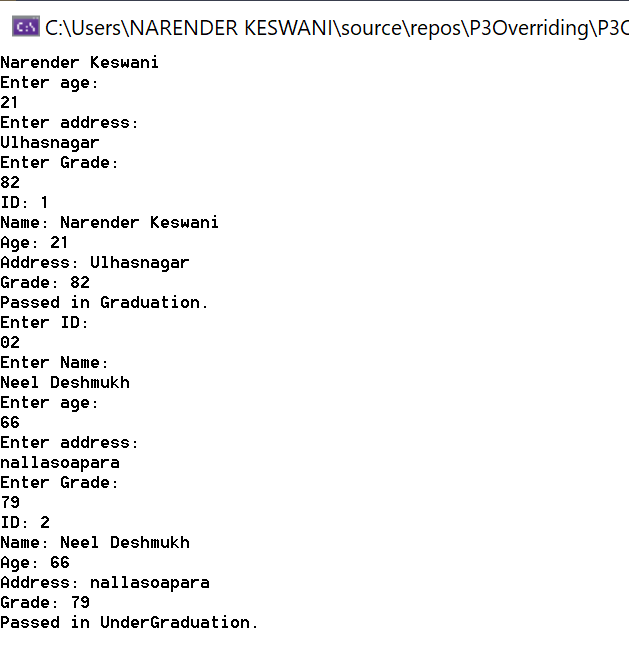
Console.ReadLine();

}

}

}

**OUTPUT:**

****

1. **Program to calculate To find the area of various shapes: Rectangle, Circle, Ellipse, Square and Triangle using abstract class and abstract method.**

**SOURCE CODE:**

using System;

namespace P3Abstract

{

public abstract class Shape

{

public double area;

public abstract void Area();

public void displayData()

{

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

}

}

public class Rectangle : Shape

{

public double len, breadth;

public void getData()

{

Console.WriteLine("Enter length of Rectangle");

len = Double.Parse(Console.ReadLine());

Console.WriteLine("Enter breadth of Rectangle");

breadth = Double.Parse(Console.ReadLine());

}

public override void Area()

{

getData();

area = len \* breadth;

displayData();

}

}

public class Circle : Shape

{

public double r;

public void getData()

{

Console.WriteLine("Enter radius of Circle");

r = Double.Parse(Console.ReadLine());

}

public override void Area()

{

getData();

area = 3.14 \* r \* r;

displayData();

}

}

public class Ellipse : Shape

{

public double a,b;

public void getData()

{

Console.WriteLine("Enter a axis of Ellipse");

a = Double.Parse(Console.ReadLine());

Console.WriteLine("Enter b axis of Ellipse");

b = Double.Parse(Console.ReadLine());

}

public override void Area()

{

getData();

area = 3.14 \* a \* b;

displayData();

}

}

public class Square : Shape

{

public double s;

public void getData()

{

Console.WriteLine("Enter side of Square");

s = Double.Parse(Console.ReadLine());

}

public override void Area()

{

getData();

area = s \* s;

displayData();

}

}

public class Triangle : Shape

{

public double h, b;

public void getData()

{

Console.WriteLine("Enter height of Triangle");

h = Double.Parse(Console.ReadLine());

Console.WriteLine("Enter breadth of Triangle");

b = Double.Parse(Console.ReadLine());

}

public override void Area()

{

getData();

area = 0.5 \* h \* b;

displayData();

}

}

class Program

{

static void Main(string[] args)

{

Rectangle r = new Rectangle();

r.Area();

Circle c = new Circle();

c.Area();

Ellipse e = new Ellipse();

e.Area();

Square s = new Square();

s.Area();

Triangle t = new Triangle();

t.Area();

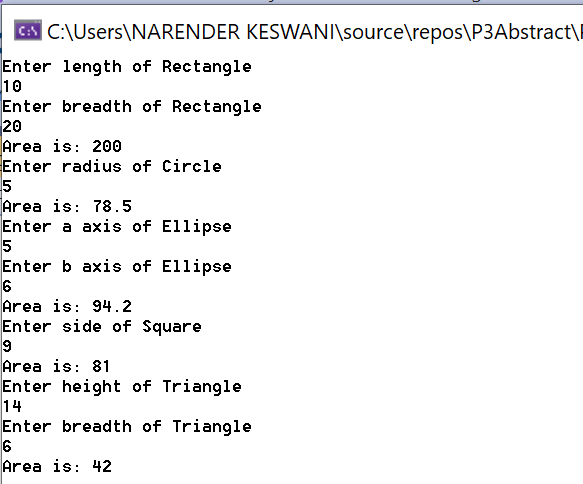
Console.ReadLine();

}

}

}

**OUTPUT:**

****

**CONCLUSION:**

From this practical, I have learned about types of inheritance and overriding in C#.