1.Student Details

class Student

{

public int StudentRollNo { get; set; }

public string StudentName { get; set; }

}

class Program

{

static void Main(string[] args)

{

Student student = new Student();//we have created an object.

Console.WriteLine("Please enter Student Roll No");

student.StudentRollNo = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please enter Student Name");

student.StudentName = Console.ReadLine();

Console.WriteLine("Student Details are : - ");

Console.WriteLine("Student Roll No. : " + student.StudentRollNo);

Console.WriteLine("Student Name : " + student.StudentName);

//Console.WritrLine();

Console.ReadLine();

}

}

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.Student get info

public class Student

{

public int StudentRollNo;

public string StudentName;

public void GetStudentData()

{

Console.WriteLine("Please enter Student Roll No");

StudentRollNo = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please enter Student Name");

StudentName = Console.ReadLine();

}

public void PrintStudentDetails()

{

Console.WriteLine("Student Details are : - ");

Console.WriteLine("Student Roll No. : " + StudentRollNo);

Console.WriteLine("Student Name : " + StudentName);

}

}

public class Program

{

public static void Main(string[] args)

{

Student student = new Student();//we have created an object.

student.GetStudentData();

student.PrintStudentDetails();

//Console.WriteLine();

Console.ReadLine();

}

}

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.Copy and Parameterised Constructor

public class Student

{

public int StudentRollNo;

public string StudentName;

public Student(Student stu) //Declaring Copy Constructor

{

StudentRollNo = stu.StudentRollNo;

StudentName = stu.StudentName;

}

public Student(int StudentRollNo, string StudentName) //parametrised constructor

{

this.StudentRollNo = StudentRollNo;

this.StudentName = StudentName;

}

public void PrintStudentDetails()

{

Console.WriteLine("Student Details are : - ");

Console.WriteLine("Student Roll No. : " + StudentRollNo );

Console.WriteLine("Student Name : " + StudentName );

}

}

class Class1

{

public static void Main()

{

Student stu1 = new Student(101,"Kalyani Kuche");

Student stu2 = new Student(stu1);

stu2.PrintStudentDetails();

Console.ReadLine();

}

}

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4.private constructor

public class counter

{

private counter()

{

}

}

public class Program

{

public static void Main()

{

Console.WriteLine("Private COnstructor");

Console.WriteLine();

}

}

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5.Inheritance

public class Employee //base class /parent class

{

public float salary = 40000;

}

public class Programmer: Employee //Derived class /child class

{

public float bonus = 10000;

}

class Class1

{

public static void Main()

{

Programmer p1 = new Programmer();

Console.WriteLine(p1.bonus);

Console.WriteLine(p1.salary);

Console.ReadLine();

}

}

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6.

class Class1

{

public class Animal //base class /parent class

{

public void eat()

{

Console.WriteLine("Dog can eat");

}

}

public class Dog: Animal //Derived class /child class

{

public void bark()

{

Console.WriteLine("Dog can Bark");

}

}

public static void Main()

{

Dog D = new Dog ();

D.eat();

D.bark();

Console.ReadLine();

}

}

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

7.Shape

public class shape

{

}

public class rectangle : shape

{

public void rectarea(int a, int b)

{

Console.WriteLine("area of rect is " + (a \* b));

}

}

public class square : rectangle

{

public void squarearea(int s)

{

Console.WriteLine("area of square is " + (s \* s));

}

}

public class Program

{

public static void Main()

{

square s1 = new square();

Console.WriteLine("calculate area for 1.rectangle\n2.area of square");

int choice = Convert.ToInt32(Console.ReadLine());

switch (choice)

{

case 1:

{

Console.WriteLine("enter length");

int a = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("enter breadth");

int b = Convert.ToInt32(Console.ReadLine());

s1.rectarea(a, b);

break;

}

case 2:

{

Console.WriteLine("enter side of square");

int s = Convert.ToInt32(Console.ReadLine());

s1.squarearea(s);

break;

}

default:

Console.WriteLine("invalid input");

break;

}

}

}

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

8.Abstract

public abstract class wholeseller

{

public abstract void Discount();

}

public class retailer : wholeseller

{

int price=0;

public void Price()

{

Console.WriteLine("enter price");

price=Convert.ToInt32(Console.ReadLine());

}

public override void Discount()

{

Console.WriteLine("enter discount");

int discount=Convert.ToInt32(Console.ReadLine());

int amt=price-discount;

Console.WriteLine("Discounted amount is "+amt);

}

}

public class Program

{

public static void Main(string[] args)

{

retailer r1 = new retailer();

r1.Price();

r1.Discount();

}

}

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

9.Person

class Class2

{

public class person

{

public int age;

public void Greet()

{

Console.WriteLine("Hello!");

}

public void SetAge(int age )

{

this.age = age;

}

}

public class student : person

{

public void study()

{

Console.WriteLine("I'm studying on screen");

}

public void ShowAge()

{

Console.WriteLine("My age is " +age+ " years old", age);

}

}

public class teacher : person

{

public void explain()

{

Console.WriteLine("I'm explaining on screen");

}

}

class program

{

public static void Main()

{

person p = new person();

p.Greet();

student s = new student();

Console.WriteLine("Enter the age : ");

int age1 = Convert.ToInt32(Console.ReadLine());

s.SetAge(age1);

s.Greet();

s.ShowAge();

s.study();

teacher t = new teacher();

Console.WriteLine("Enter the age of the person: ");

int age2 = Convert.ToInt32(Console.ReadLine());

t.SetAge(age2);

t.Greet();

t.explain();

Console.ReadLine();

}

}

}

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

10.Photo book

public class PhotoBookTest

{

public static void Main()

{

PhotoBook myAlbum1 = new PhotoBook();

Console.WriteLine(myAlbum1.GetNumberPages());

PhotoBook myAlbum2 = new PhotoBook(24);

Console.WriteLine(myAlbum2.GetNumberPages());

SuperPhotoBook myBigPhotoAlbum1 = new SuperPhotoBook();

Console.WriteLine(myBigPhotoAlbum1.GetNumberPages());

}

}

public class PhotoBook

{

protected int numPages;

public PhotoBook()

{

numPages = 16;

}

public PhotoBook(int numPages)

{

this.numPages = numPages;

}

public int GetNumberPages()

{

return numPages;

}

}

public class SuperPhotoBook : PhotoBook

{

public SuperPhotoBook()

{

numPages = 64;

}

}