**Q 1:** Create a base class called Shape with a method CalculateArea (). Create two derived classes Rectangle and Circle that inherit from Shape and override the CalculateArea () method to calculate the area of a rectangle and a circle respectively.

Write a C# program to demonstrate the use of inheritance.

**Solution:**

class Shape

{

public virtual void CalculateArea()

{

Console.WriteLine("Calculating area of the shape...");

}

}

class Rectangle : Shape

{

public override void CalculateArea()

{

Console.WriteLine("Calculating area of rectangle...");

// Calculate area of rectangle here

}

}

class Circle : Shape

{

public override void CalculateArea()

{

Console.WriteLine("Calculating area of circle...");

// Calculate area of circle here

}

}

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

class Program

{

static void Main()

{

Shape shape1 = new Rectangle();

shape1.CalculateArea();

Shape shape2 = new Circle();

shape2.CalculateArea();

}

}

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

**Question 2:** Create a base class called Animal with a method MakeSound(). Create two derived classes Dog and Cat that inherit from Animal and override the MakeSound() method to make the sound of a dog and a cat respectively. Write a C# program to demonstrate the use of inheritance.

**Solution**:

class Animal

{

public virtual void MakeSound()

{

Console.WriteLine("Making animal sound...");

}

}

class Dog : Animal

{

public override void MakeSound()

{

Console.WriteLine("Woof! Woof!");

}

}

class Cat : Animal

{

public override void MakeSound()

{

Console.WriteLine("Meow! Meow!");

}

}

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

class Program

{

static void Main()

{

Animal animal1 = new Dog();

animal1.MakeSound();

Animal animal2 = new Cat();

animal2.MakeSound();

}

}

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

**Question 3:** Create a base class called Vehicle with a virtual method StartEngine(). Create two derived classes Car and Motorcycle that inherit from Vehicle and override the StartEngine() method to start the engine of a car and a motorcycle respectively. Write a C# program to demonstrate method overriding.

**Solution**:

class Vehicle

{

public virtual void StartEngine()

{

Console.WriteLine("Starting the engine of the vehicle...");

}

}

class Car : Vehicle

{

public override void StartEngine()

{

Console.WriteLine("Starting the engine of the car...");

}

}

class Motorcycle : Vehicle

{

public override void StartEngine()

{

Console.WriteLine("Starting the engine of the motorcycle...");

}

}

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

class Program

{

static void Main()

{

Vehicle vehicle1 = new Car();

vehicle1.StartEngine();

Vehicle vehicle2 = new Motorcycle();

vehicle2.StartEngine();

}

}

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

**Question 4:** Create a base class called Person with properties Name and Age. Create two derived classes Student and Teacher that inherit from Person and have additional properties Grade and Subject respectively. Write a C# program to demonstrate inheritance and access the properties of the derived classes.

**Solution**:

class Person

{

public string Name { get; set; }

public int Age { get; set; }

}

class Student : Person

{

public int Grade { get; set; }

}

class Teacher : Person

{

public string Subject { get; set; }

}

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

class Program

{

static void Main()

{

Student student = new Student()

{

Name = "John",

Age = 18,

Grade = 12

};

Teacher teacher = new Teacher()

{

Name = "Jane",

Age = 30,

Subject = "Math"

};

Console.WriteLine("Student Information:");

Console.WriteLine("Name: {0}", student.Name);

Console.WriteLine("Age: {0}", student.Age);

Console.WriteLine("Grade: {0}", student.Grade);

Console.WriteLine("Teacher Information:");

Console.WriteLine("Name: {0}", teacher.Name);

Console.WriteLine("Age: {0}", teacher.Age);

Console.WriteLine("Subject: {0}", teacher.Subject);

}

}

**Question 5:** Create a base class called **Employee** with properties **Name** and **Salary**. Create a derived class **Manager** that inherits from **Employee** and adds a property **Department**. Write a C# program to demonstrate inheritance and access the properties of the derived.

Solution:

class Employee

{

public string Name { get; set; }

public double Salary { get; set; }

}

class Manager : Employee

{

public string Department { get; set; }

}

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

class Program

{

static void Main()

{

Manager manager = new Manager()

{

Name = "John",

Salary = 50000,

Department = "Sales"

};

Console.WriteLine("Manager Information:");

Console.WriteLine("Name: {0}", manager.Name);

Console.WriteLine("Salary: {0}", manager.Salary);

Console.WriteLine("Department: {0}", manager.Department);

}

}

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