Practical – 3
Inheritance and Interfaces

1. Write a program to implement inheritance from following figure.

```
using System;
namespace Labwork3
{
   abstract public class Car
   {
      public int Carno { get; set; }
      public string Name { get; set; }

      public abstract void GetCarInfo();
   }

   public interface IRate
   {
      double RateKm { get; }
      double RateDay { get; }
      double GetDays();
      double GetKm();
   }
   public class Customer : Car, IRate
   {
      public double RateKm { get; } = 7.5;
      public double RateDay { get; } = 500;

      public double GetDays()
      {
         return 5;
      }
      public double GetKm()
      {
         return 300;
      }
      public double CalculateRate()
      {
         double totalAmount = (GetDays() * RateDay) + (GetKm() * RateKm);
         return totalAmount;
      }
      public override void GetCarInfo()
      {
         Console.WriteLine("Car Number: " + Carno);
```

Practical – 3
Inheritance and Interfaces

```
    Console.WriteLine("Car Name: " + Name);
        }
    }

    class Pro1
    {
        static void Main(string[] args)
        {
            Customer customer = new Customer();
            customer.Carno = 123;
            customer.Name = "BMW";
            customer.GetCarInfo();

            double totalAmount = customer.CalculateRate();
            Console.WriteLine("Total Amount: Rs." + totalAmount);
            Console.ReadLine();
        }
    }
}
```

- **Output**

```
Car Number: 123
Car Name: BMW
Total Amount: Rs.4750
```

2. Write a program to implement single inheritance from following figure. Accept and display data for one table.

```
using System;
namespace Labwork3
{
    public class Furniture
    {
        public string material;
        public double price;
    }
    public class Table:Furniture
    {
        public double Height;
```

Practical – 3
Inheritance and Interfaces

```
 public double surface_area;

   }
   class pro2
   {
      static void Main(string[] args)
      {
         Table tobj = new Table();
         tobj.material = "Wooden";
         tobj.price = 45000;
         tobj.Height = 4.5;
         tobj.surface_area = 5.6;
         Console.WriteLine("Table information:");
         Console.WriteLine("material=" + tobj.material);
         Console.WriteLine("price=" + tobj.price);
         Console.WriteLine("height=" + tobj.Height);
         Console.WriteLine("surface area=" + tobj.surface_area);
         Console.ReadLine();
      }
   }
}
```

- **Output**

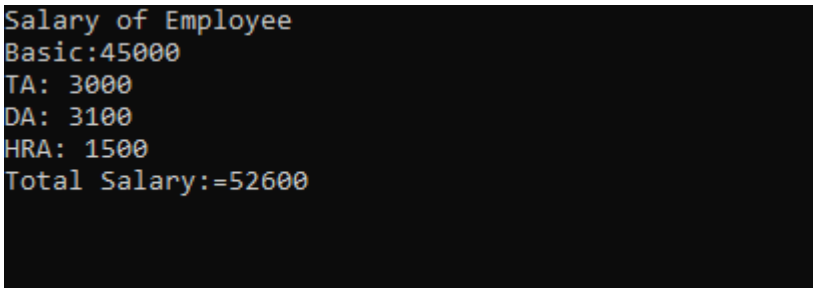
3. Define a class “salary‟ which will contain member variable Basic, TA, DA, HRA. Write a program using Constructor with default values for DA and HRA and calculate the salary of employee.

```
using System;
namespace Labwork3
{
   public class salary
   {
     public  double Basic, TA, DA, HRA;
      public double tBasic;
```

Practical – 3
Inheritance and Interfaces

```
        public salary()
        {
            DA = 3100;
            HRA = 1500;
        }
        public void calcuatesalary()
        {
            tBasic = Basic + TA + DA + HRA;
            Console.WriteLine("Salary of Employee");
            Console.WriteLine("Basic:" + Basic);
            Console.WriteLine("TA: " + TA);
            Console.WriteLine("DA: " + DA);
            Console.WriteLine("HRA: " + HRA);
            Console.WriteLine("Total Salary:=" + tBasic);
        }
    }
    class pro3:salary
    {
        static void Main(string[] args)
        {
            salary sobj = new salary();
            sobj.TA = 3000;
            sobj.Basic = 45000;
            sobj.calcuatesalary();
            Console.ReadLine();
        }
    }
}
```

- **Output**



Practical – 3
Inheritance and Interfaces

4. Program to implement the following multiple inheritance using interface.

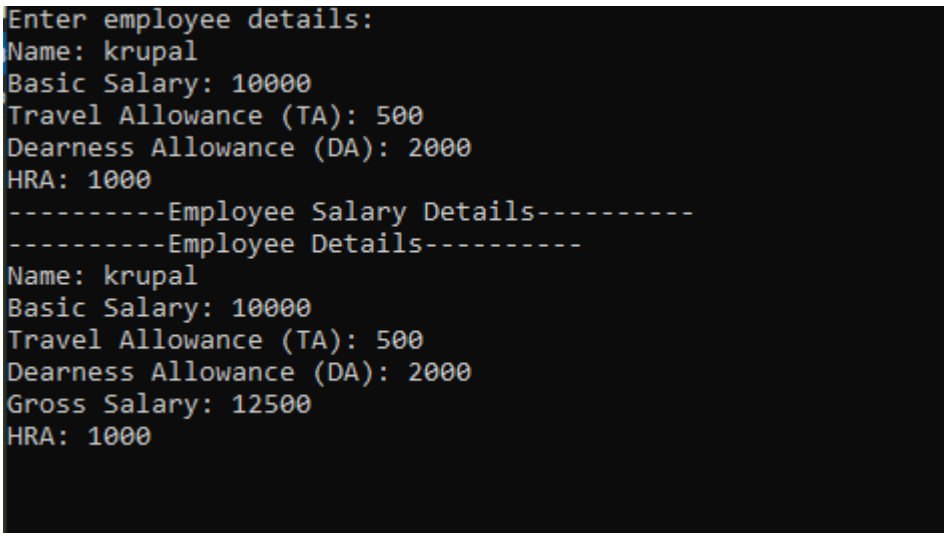
```
using System;
namespace Labwork3
{
   public interface Gross
   {
      double TA { get; set; }
      double DA { get; set; }
      double Gross_sal();
   }

   public class Employee1 : Gross
   {
      public string Name;
      public double BasicSalary;

      public Employee1(string name, double basicSalary, double ta, double da)
      {
         Name = name;
         BasicSalary = basicSalary;
         TA = ta;
         DA = da;
      }
      public double TA { get; set; }
      public double DA { get; set; }

      public double Gross_sal()
      {
         return BasicSalary + TA + DA;
      }

      public void basic_sal()
      {
         Console.WriteLine("----------Employee Details----------");
         Console.WriteLine("Name: " + Name);
         Console.WriteLine("Basic Salary: " + BasicSalary);
      }
   }

   public class Salary : Employee1
   {
      public double HRA;
```

Practical – 3
Inheritance and Interfaces

```
    public Salary(string name, double basicSalary, double ta, double da, double hra)
        : base(name, basicSalary, ta, da)
    {
        HRA = hra;
    }
    public double CalculateHRA()
    {
        return HRA;
    }
    public void Disp_sal()
    {
        Console.WriteLine("----------Employee Salary Details----------");
        basic_sal();
        Console.WriteLine("Travel Allowance (TA): " + TA);
        Console.WriteLine("Dearness Allowance (DA): " + DA);
        Console.WriteLine("Gross Salary: " + Gross_sal());
        Console.WriteLine("HRA: " + CalculateHRA());
    }
}

class pro4
{
    static void Main(string[] args)
    {
        Console.WriteLine("Enter employee details:");
        Console.Write("Name: ");
        string name = Console.ReadLine();
        Console.Write("Basic Salary: ");
        double basicSalary = Convert.ToDouble(Console.ReadLine());
        Console.Write("Travel Allowance (TA): ");
        double ta = Convert.ToDouble(Console.ReadLine());
        Console.Write("Dearness Allowance (DA): ");
        double da = Convert.ToDouble(Console.ReadLine());
        Console.Write("HRA: ");
        double hra = Convert.ToDouble(Console.ReadLine());

        Salary empSalary = new Salary(name, basicSalary, ta, da, hra);
        empSalary.Disp_sal();

        Console.ReadLine();
    }
}
}
```

Practical – 3
Inheritance and Interfaces

- **Output**



5. Write a program for above class hierarchy for the Employee where the base class is Employee and derived class and Programmer and Manager. Here make display function virtual which is common for all and which will display information of Programmer and Manager interactively.

```
using System;
namespace Labwork3
{
   public class Employee
   {
      public string empno;
      public string empname;
      public long mobile;
      public double salary;
      public Employee(string empno, string empname, long mobile, double salary)
      {
         this.empno = empno;
         this.empname = empname;
         this.mobile = mobile;
         this.salary = salary;
      }
      public virtual void Display()
      {
         Console.WriteLine("-----Employee details-----");
         Console.WriteLine("NO: " + empno);
         Console.WriteLine("Name: " + empname);
         Console.WriteLine("Mobile: " + mobile);
```

Practical – 3
Inheritance and Interfaces

```
            Console.WriteLine("Salary: " + salary);
        }
    }
    public class Programmer : Employee
    {
        public Programmer(string empno, string empname, long mobile, double salary)
            : base(empno, empname, mobile, salary)
        {
        }
    }
    public class Manager : Employee
    {
        public Manager(string empno, string empname, long mobile, double salary)
            : base(empno, empname, mobile, salary)
        {
        }
    }
    class pro5
    {
        static void Main(string[] args)
        {
            Programmer pobj = new Programmer("e02", "krupal pandya", 9408520471, 56000);
            Manager mobj = new Manager("e05", "dev", 9493940000, 70000);

            Console.WriteLine("Programmer Information:");
            pobj.Display();

            Console.WriteLine("Manager Information:");
            mobj.Display();

            Console.ReadLine();
        }
    }
}
```

- **Output**

```
Programmer Information:
-----Employee details-----
NO: e02
Name: krupal pandya
Mobile: 9408520471
Salary: 56000
Manager Information:
-----Employee details-----
NO: e05
Name: dev
Mobile: 9493940000
Salary: 70000
```

Practical – 3
Inheritance and Interfaces

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

```
using System;
namespace Labwork3
{
   public class student
   {
      public int rollno;
      public string name;
      public student(int rollno,string name)
      {
         this.rollno = rollno;
         this.name = name;
      }
   }
   public class Test : student
   {
      public int marks1, marks2;
      public Test(int rollno,string name,int marks1,int marks2):base(rollno,name)
      {

         this.marks1 = marks1;
         this.marks2 = marks2;
      }
   }
   class result : Test
   {
      int total = 0;
      public result(int rollno, string name, int marks1, int marks2)
         : base(rollno, name, marks1, marks2)
      {
         total = marks1 + marks2;
      }
      public void display()
      {
         Console.WriteLine("----------Student Details----------");
         Console.WriteLine("Roll No:" + rollno);
         Console.WriteLine("Name:" + name);
         Console.WriteLine("Marks1:" + marks1);
         Console.WriteLine("Marks2:" + marks2);
         Console.WriteLine("Total:" + total);
      }
      public static void Main(string[] args)
```

Practical – 3
Inheritance and Interfaces

```
    {
        Console.WriteLine("enter student details:");
        Console.WriteLine("Enter student rollno:");
        int rollno = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter student name:");
        string name = Console.ReadLine();
        Console.WriteLine("Enter marks of two subjects:");
        int marks1= Convert.ToInt32(Console.ReadLine());
        int marks2= Convert.ToInt32(Console.ReadLine());
        result robj = new result(rollno,name,marks1,marks2);
        robj.display();
        Console.ReadLine();

    }
  }
}
```

- **Output**