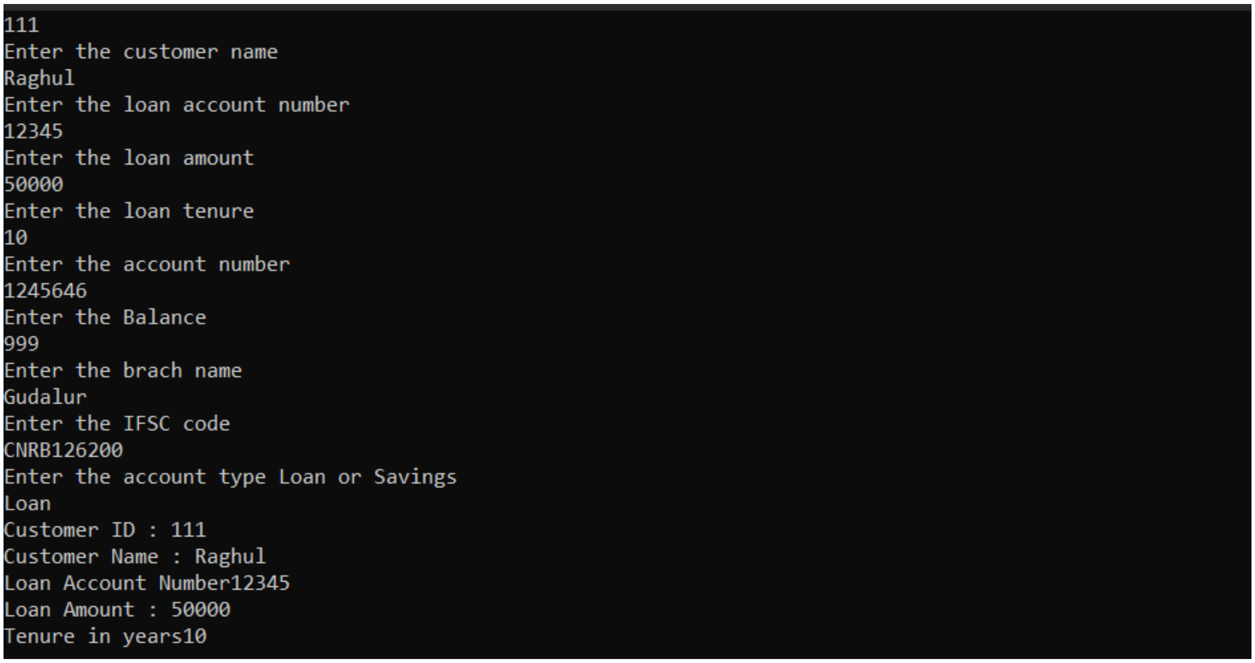
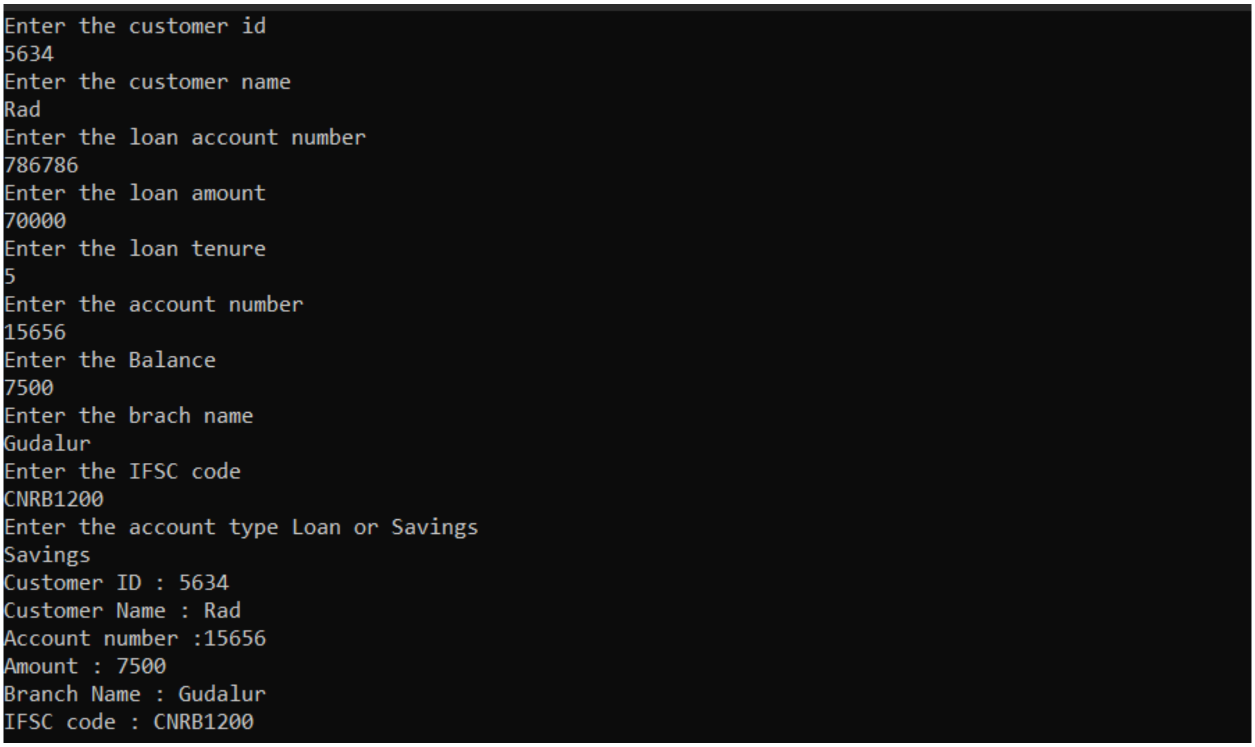
```
using Basic_Program;
using System.Net.Http.Headers;

class Demo1
{
    public static void Main(string[] args)
    {

             Console.WriteLine("Enter the customer id");
        int custid = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Enter the customer name");
        string custname = Console.ReadLine();

        Console.WriteLine("Enter the loan account number");
        int loanAccNo = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Enter the loan amount");
        int loanAmount = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Enter the loan tenure");
        int loanTenureinyrs = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Enter the account number");
        int accno = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Enter the Balance");
        double amount = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Enter the brach name");
        string branchname = Console.ReadLine();

        Console.WriteLine("Enter the IFSC code");
        string ifsc = Console.ReadLine();


        Customer customer = new Customer(custid, custname);
        LoanAccount loanAccount = new LoanAccount(loanAccNo, loanAmount, loanTenureinyrs, custid,
custname);
        SavingsAccount savingsAccount = new SavingsAccount(accno, amount, branchname, ifsc, custid,
custname);


        Console.WriteLine("Enter the account type Loan or Savings" );
        string ch = Console.ReadLine();
        switch (ch)
        {
            case "Loan" or "loan":
                loanAccount.CustDetails();
                break;
            case "Savings" or "savings":
                savingsAccount.CustDetails();
                break;
            default:
                Console.WriteLine("Enter Laon or savings");
                break;
        }
    }
}



using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Basic_Program
{
    internal class Customer
    {
        private int custid;
        private string custname;

        public Customer(int custid, string custname)
        {
              this.Custid = custid;
              this.Custname = custname;
        }

        public int Custid { get => custid; set => custid = value; }
        public string Custname { get => custname; set => custname = value; }

        public void CustDetails()
        {
            Console.WriteLine(this.Custid);
            Console.WriteLine(this.Custname);
        }
    }
}



using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Basic_Program
{
    internal class LoanAccount:Customer
    {
        private int loanAccNo;
        private int loanAmount;
        private int loanTenureinyrs;
        public LoanAccount(int loanAccNo, int loanAmount, int loanTenureinyrs, int custid, string
custname):base(custid, custname)
        {
            this.LoanAccNo = loanAccNo;
            this.LoanAmount = loanAmount;
            this.LoanTenureinyrs = loanTenureinyrs;
        }

        public int LoanAccNo { get => loanAccNo; set => loanAccNo = value; }
        public int LoanAmount { get => loanAmount; set => loanAmount = value; }
        public int LoanTenureinyrs { get => loanTenureinyrs; set => loanTenureinyrs = value; }

        public void CustDetails()
        {
            Console.WriteLine("Customer ID : " + this.Custid);
            Console.WriteLine("Customer Name : "+ this.Custname);
            Console.WriteLine("Loan Account Number"+ this.LoanAccNo);
            Console.WriteLine("Loan Amount : " + this.LoanAmount);
            Console.WriteLine("Tenure in years" + this.LoanTenureinyrs);
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Basic_Program
{
    internal class SavingsAccount:Customer
    {
        private int accno;
        private double amount;
        private string branchname;
        private string ifsc;

        public SavingsAccount(int accno, double amount, string branchname, string ifsc, int custid, string custname) :base(custid, custname)
        {
            this.Accno = accno;
            this.Amount = amount;
            this.Branchname = branchname;
            this.Ifsc = ifsc;
        }

        public int Accno { get => accno; set => accno = value; }
        public double Amount { get => amount; set => amount = value; }
        public string Branchname { get => branchname; set => branchname = value; }
        public string Ifsc { get => ifsc; set => ifsc = value; }


        public void CustDetails()
        {
            Console.WriteLine("Customer ID : " + this.Custid);
            Console.WriteLine("Customer Name : " + this.Custname);
            Console.WriteLine("Account number :" + this.Accno);
            Console.WriteLine("Amount :" + this.Amount);
            Console.WriteLine("Branch Name : " + this.Branchname);
            Console.WriteLine("IFSC code : " + this.Ifsc);
        }
    }
}
```



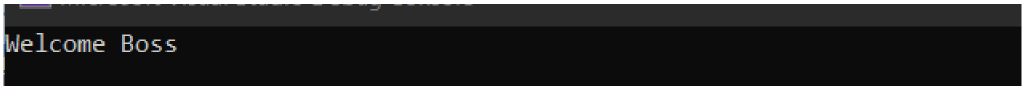
```
111
Enter the customer name
Raghul
Enter the loan account number
12345
Enter the loan amount
50000
Enter the loan tenure
10
Enter the account number
1245646
Enter the Balance
999
Enter the brach name
Gudalur
Enter the IFSC code
CNRB126200
Enter the account type Loan or Savings
Loan
Customer ID : 111
Customer Name : Raghul
Loan Account Number12345
Loan Amount : 50000
Tenure in years10
```



```
Enter the customer id
5634
Enter the customer name
Rad
Enter the loan account number
786786
Enter the loan amount
70000
Enter the loan tenure
5
Enter the account number
15656
Enter the Balance
7500
Enter the brach name
Gudalur
Enter the IFSC code
CNRB1200
Enter the account type Loan or Savings
Savings
Customer ID : 5634
Customer Name : Rad
Account number :15656
Amount : 7500
Branch Name : Gudalur
IFSC code : CNRB1200
```

2. Work with concept of abstract class with own implementation.

```
using Basic_Program;
using System.Net.Http.Headers;

class Demo1
{
    public static void Main(string[] args)
    {

              DerivedAbstractClass derivedAbstractClass = new DerivedAbstractClass();
        derivedAbstractClass.display();
    }
}



using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Basic_Program
{
    internal abstract class AbstractClass
    {
        public abstract void display();
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Basic_Program
{
    internal class DerivedAbstractClass:AbstractClass
    {
        public override void display()
        {
            Console.WriteLine("Welcome Boss");
        }
    }
}
```



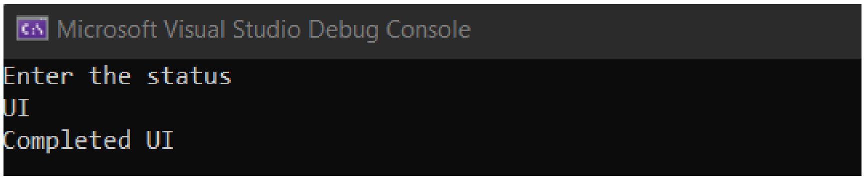
```
Welcome Boss
```

3. Work with concept of interface

```
using Basic_Program;
using System.Net.Http.Headers;

class Demo1
{
    public static void Main(string[] args)
    {

              UI ui = new UI();
        DBdesign dBdesign = new DBdesign();
        Console.WriteLine("Enter the status");
        string status = Console.ReadLine();
        switch(status)
        {
            case "UI":
                Console.WriteLine(ui.Progress());
                break;

            case "DBdesign":
                Console.WriteLine(dBdesign.Progress());
                break;
            default:
                Console.WriteLine("Enter UI or DBdesign");
                break;
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Basic_Program
{
    internal class UI:IProject
    {
        public string Progress()
        {
            return "Completed UI";
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Basic_Program
{
    internal class DBdesign:IProject
    {
        public string Progress()
        {
            return "Completed DB Design";
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Basic_Program
{
    internal interface IProject
    {
        string Progress();
    }
}
```



Microsoft Visual Studio Debug Console

```
Enter the status
UI
Completed UI
```

4. Implement multiple inheritance with abstract class and interface.

```
namespace basicprgm1
{
    internal class multipleclass
    {
        public void display()
        {
            Console.WriteLine("Base Class 1");
        }
    }
}

namespace basicprgm1
{
    internal interface multipleinterface
    {
        public void display1();
    }
}


namespace basicprgm1
{
    internal class derivedclass : multipleclass, multipleinterface
    {
        public derivedclass() : base()
        { }
        public void derived()
        {
            Console.WriteLine("Derived Class");
        }
        public void display1()
        {
            Console.WriteLine("Base class 2");
        }
    }
}


derivedclass d = new derivedclass();
d.display();
```

1.

```
namespace Assign3
{
    internal class Program
    {
        public static void Main(string[] args)
        {
            Console.Write("Enter limit: ");
            int limit = Convert.ToInt32(Console.ReadLine());

            int[] array1 = new int[limit];

            for (int i = 0; i < limit; i++)
            {
                array1[i] = Convert.ToInt32(Console.ReadLine());
            }

            for (int i = 0; i < limit; i++)
            {
                for (int j = i + 1; j < limit; j++)
                {
                    if (array1[i] > array1[j])
                    {
                        int temp = array1[i];
                        array1[i] = array1[j];
                        array1[j] = temp;
                    }
                }
            }
            Console.WriteLine("-------------------------------");
            Console.WriteLine("Ascending Order Sort is displayed Below");
            Console.Write(String.Join(",", array1));
        }
    }
}
```

2.Just implement class and use getter and setter in it

```
namespace Assign3
{
    internal class Array
    {
        private int id;
        private string name;
        private int age;
        private string addr;
        public Array(int id, string name, int age, string addr)
        {
            this.id = id;
            this.name = name;
            this.age = age;
            this.addr = addr;
        }

        public int Id { get => id; set => id = value; }
        public string Name { get => name; set => name = value; }
        public int Age { get => age; set => age = value; }
        public string Addr { get => addr; set => addr = value; }


        public void display()
        {
            Console.WriteLine(this.id);
            Console.WriteLine(this.name);
            Console.WriteLine(this.age);
            Console.WriteLine(this.addr);
        }
    }
    internal class Program
    {
        public static void Main(string[] args)
        {
            Console.Write("Enter id: ");
            int id = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter Name: ");
            string name = Console.ReadLine();

            Console.Write("Enter Age: ");
            int age = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter addr: ");
            string addr = Console.ReadLine();

            Console.WriteLine("----------------------------------------------");

            Array array = new Array(id, name, age, addr);
            array.display();
        }
    }
}
```

3.

```
    public void jagged()
{
    int i, j;
    int[][] jaggedArray = new int[4][];
    jaggedArray[0] = new int[] { 1, 2, 3, 4, 5 };
    jaggedArray[1] = new int[] { 40, 50, 11, 4 };
    jaggedArray[2] = new int[] { 55, 17 };
    jaggedArray[3] = new int[4];

    Console.WriteLine(jaggedArray[0][2]);

    for (i = 0; i < jaggedArray[3].Length; i++)
    {
        jaggedArray[3][i] = Convert.ToInt32(Console.ReadLine());
    }


    int[][,] jaggy = new int[3][,];
    jaggy[0] = new int[5, 4];
    jaggy[1] = new int[6, 5];
    jaggy[2] = new int[7, 6];

    for (i = 0; i < 5; i++)
    {
        for (j = 0; j < 4; j++)
        {
            jaggy[0][i, j] = Convert.ToInt32(Console.ReadLine());
        }
    }

    for (i = 0; i < 5; i++)
    {
        for (j = 0; j < 4; j++)
        {
            Console.Write(jaggy[0][i, j]);
        }
        Console.WriteLine();
    }
}
```