19-39587-1

Annanto,sijan shariar (N)

```
using System;

namespace Interface1

{
   interface calculator
   {

   }
   class BasicCalculator : calculator
   {
      int x1;
      int y1;
      public int sum(int x, int y)
      {
         x1 = x;
         y1 = y;
         return x + y;
      }
      public int sub(int x, int y)
      {
         x1 = x;
         y1 = y;
```

```
        return x - y;
    }
    public int multiplication(int x, int y)
    {
        x1 = x;
        y1 = y;
        return x*y;
    }
    public int division(int x, int y)
    {
        x1 = x;
        y1 = y;
        return x / y;
    }

}
 class ScientificCalculator : calculator
{
    int x1;
    int y1;

    public int sum(int x, int y)
    {
        x1 = x;
        y1 = y;
        return x + y;
    }
    public int sub(int x, int y)
    {
```

```
    x1 = x;
    y1 = y;
    return x - y;
}
public int multiplication(int x, int y)
{
    x1 = x;
    y1 = y;
    return x * y;
}
public int division(int x, int y)
{
    x1 = x;
    y1 = y;
    return x / y;
}
public double XtoY(int x, int y)
{
    x1 = x;
    y1 = y;
  //  double pob = Convert.ToInt32(Console.ReadLine());
     double pob = Math.Pow(2, 6);
    return pob;
}
public double Exponential(int x)
{
    x1 = x;
    double exp = Math.Exp(10);
    return exp;
```

```
    }
  }


  class Program
  {
    static void Main(string[] args)
    {
      Console.WriteLine("Basic Calculator ..... ");
      BasicCalculator c1 = new BasicCalculator();
      Console.WriteLine("\nSum : " + c1.sum(4,2));
      Console.WriteLine("Sub : " + c1.sub(4,2));
      Console.WriteLine("Multiplication : "  + c1.multiplication(4,2));
      Console.WriteLine("Division : " + c1.division(4,2));

      Console.WriteLine("\nScientific Calculator ..... ");
      ScientificCalculator c2 = new ScientificCalculator();
      Console.WriteLine("\nSum : " + c2.sum(2, 6));
      Console.WriteLine("Sub : " + c2.sub(2, 6));
      Console.WriteLine("Multiplication : " + c2.multiplication(2, 6));
      Console.WriteLine("Division : " +c2.division(4, 2));
      Console.WriteLine("X to Y : " +c2.XtoY(2,6));
      Console.WriteLine("Exponential : " +c2.Exponential(10));

    }
  }
}
```

BasicBankingInterface

```
using System;
using System.Collections.Generic;
using System.Text;

namespace AssignmentMidTask2
{
   interface BasicBankingInterface
   {
      bool deposit(int amount);
      bool withdraw(int amount);

   }
}
```

```
using System;
using System.Collections.Generic;
using System.Text;

namespace AssignmentMidTask2
{
   class BankAccount : BasicBankingInterface
   {
```

```
protected string accId;
protected static int count = 0;
protected int balance;
public string accountType;
public person user;
private static person[] customers = new person[1000];
protected BankAccount bankAccount;




public BankAccount()
{

}

public BankAccount( person user)
{
   this.user = user;
   addCustomers(user);

}

public BankAccount(string accountType)
{
   this.accountType = accountType;
}
```

```
public int Balance
{

   get
   {
      return balance;
   }




}

public person User
{
   get
   {
      return user;
   }
}









public virtual bool deposit(int amount) {

   balance = balance + amount;
```

```
    Console.WriteLine("Deposit of {0}Tk is Successful", amount);
    return true;
}

public virtual bool withdraw(int amount)
{

    bool success = false;

    if (amount <= balance)
    {
        balance = balance - amount;
        success = true;
    }
    else
    {
        Console.WriteLine("Not Enough Balance!!");

    }
    return success;
}

public void addCustomers(person customer)
{
```

```
    for (int i = 0; i < customers.Length; i++)
    {

        if (customers[i]== customer)
        {
            customers[i]=null;

        }


    }
    for (int i = 0; i < customers.Length; i++)
    {
        if (customers[i] == null)
        {
            customers[i] = customer;
            break;
        }


    }
}


internal void showAllCustomers()
{
    for (int i = 0; i < customers.Length; i++)
    {
        if (customers[i] != null)
```

```
            {

                bankAccount=customers[i].getAccount();
                bankAccount.showCustomerInfo();



            }

          }

       }

       internal virtual void showCustomerInfo()
       {
          Console.WriteLine("This Is A Virtual Method");

       }



    }
}
```

```
using System;
using System.Collections.Generic;
using System.Text;

namespace AssignmentMidTask2
```

```
{
    class CurrentAccount:BankAccount
    {

        public CurrentAccount()
        {
            Console.WriteLine("Current account is created");
        }



        public CurrentAccount(person user):base(user)
        {

            count++;
            base.user = user;
            accountType = "Current Account";
            accId = "Acc-ca" + Convert.ToString(count);
            Console.WriteLine("\nCurrent account is created for {0} with Account Id: {1}",user.Name,accId);

        }

        public CurrentAccount(BankAccount previousAccount,person user) : base(user)
        {
            Console.WriteLine("\n----->Account Type Changed for {0}", previousAccount.user.Name);
            count++;
            balance = previousAccount.Balance;
            base.user = user;
            accountType = "Current Account";
            accId = "Acc-ca" + Convert.ToString(count);
```

```
    Console.WriteLine("\nCurrent account is created for {0} with Account Id: {1}", user.Name, accId);

}


public override bool deposit(int amount)
{

    balance = balance + amount;
    Console.WriteLine("\nCurrent Account: Deposit of {0}Tk is Successful", amount);
    return true;
}

public override bool withdraw(int amount)
{
    Console.WriteLine("\n>>>>>>>>>Trying To withdraw from account Current account>>>>>>>>>");
    bool success = false;

    if (amount <= balance)
    {
        balance = balance - amount;
        success = true;
    }
    else
    {
        Console.WriteLine("Not Enough Balance!!");

    }
    return success;
```

```
        }

        internal override void showCustomerInfo()
        {
            Console.WriteLine("\n----Account Holder Information----\n");
            Console.WriteLine("Account ID: {0}", accId);
            user.showInfo();
            Console.WriteLine("Account Type: {0}", accountType);
            Console.WriteLine("Balance: {0}Tk", balance);

        }

    }
}
```

```
using System;
using System.Collections.Generic;
using System.Text;

namespace AssignmentMidTask2
{
   class OverdraftAccount:BankAccount
   {
      private int overdraftLimitAmount;



      public OverdraftAccount(person user,int overdraftLimitAmount):base(user)
      {
         count++;
         base.user = user;
         accountType = "Overdraft Account";
         this.overdraftLimitAmount = overdraftLimitAmount;
         accId = "Acc-oda" + Convert.ToString(count);
         Console.WriteLine("\nOverdraft account is created for {0} with Account Id: {1}", user.Name,
accId);
      }



      public OverdraftAccount(BankAccount previousAccount,person user, int overdraftLimitAmount) :
base(user)
      {
         Console.WriteLine("\n----->Account Type Changed for {0}", previousAccount.user.Name);
         count++;
```

```
        balance=previousAccount.Balance;
        base.user = user;
        accountType = "Overdraft Account";
        this.overdraftLimitAmount = overdraftLimitAmount;
        accId = "Acc-oda" + Convert.ToString(count);
        Console.WriteLine("\nOverdraft account is created for {0} with Account Id: {1}", user.Name,
accId);
    }

    public override bool deposit(int amount)
    {

        balance = balance + amount;
        Console.WriteLine("\nOverdraft Account: Deposit of {0}Tk is Successful", amount);
        return true;
    }




    public override bool withdraw(int amount)
    {
        Console.WriteLine("\n>>>>>>>>>>Trying To withdraw from account Overdraft
account>>>>>>>>>>");
        bool success = false;

        if (amount <= balance)
        {
            balance = balance - amount;
            success = true;
```

```
    }
    if(amount>balance && balance!=0 && amount<=(balance+overdraftLimitAmount))
    {
        int temp = amount - balance;
        amount = amount - temp;
        balance = balance - amount;
        amount = temp;



    }
    if (balance==0 && amount<=overdraftLimitAmount)
    {
        Console.WriteLine("\n****Note: Withdrawing From Overdraft Limit amount");

        overdraftLimitAmount = overdraftLimitAmount - amount;
        success = true;
    }
    else
    {
        Console.WriteLine("\n****Not Suffient balance>>>Overdraft Limit Amount Is reached!!");

    }
    return success;
}



internal override void showCustomerInfo()
{
    Console.WriteLine("\n----Account Holder Information----\n");
```

```
            Console.WriteLine("Account ID: {0}", accId);
            user.showInfo();
            Console.WriteLine("Account Type: {0}", accountType);
            Console.WriteLine("Balance: {0}Tk", balance);
            Console.WriteLine("Overdraft Limit Amount: {0}Tk", overdraftLimitAmount);

        }

    }
}
```

```
using System;
using System.Collections.Generic;
using System.Text;

namespace AssignmentMidTask2
{
    class SavingsAccount:BankAccount
    {
        private int withdrawableAmount;
```

```
public SavingsAccount()
{
   Console.WriteLine("Savings account is created");
}

public SavingsAccount(person user):base(user)
{
   count++;
   base.user = user;
   accountType = "Savings Account";
   accId = "Acc-sa" + Convert.ToString(count);
   Console.WriteLine("\nSavings account is created for {0} with Account Id: {1}", user.Name, accId);
}


public SavingsAccount(BankAccount previousAccount,person user) : base(user)
{
   Console.WriteLine("\n----->Account Type Changed for {0}", previousAccount.user.Name);
   count++;
   balance = previousAccount.Balance;
   base.user = user;
   accountType = "Savings Account";
   accId = "Acc-sa" + Convert.ToString(count);
   Console.WriteLine("\nSavings account is created for {0} with Account Id: {1}", user.Name, accId);
}


public override bool deposit(int amount)
{
```

```
    balance = balance + amount;
    Console.WriteLine("\nSavings Account: Deposit of {0}Tk is Successful", amount);
    return true;
}

public override bool withdraw(int amount)
{
    Console.WriteLine("\n>>>>>>>>>Trying To withdraw from account savings account>>>>>>>>>");
    bool success=false;
    withdrawableAmount = (balance * 80) / 100;

    if (amount <= withdrawableAmount)
    {
        balance = balance - amount;
        success = true;
    }
    else
    {
        Console.WriteLine("\n****Not Enough Balance In Savings Account!");
    }
    return success;

}

internal override void showCustomerInfo()
{
    Console.WriteLine("\n----Account Holder Information----\n");
    Console.WriteLine("Account ID: {0}", accId);
```

```
            user.showInfo();
            Console.WriteLine("Account Type: {0}",accountType);
            Console.WriteLine("Balance: {0}Tk",balance);

        }



    }
}
```

Person class

```
using System;
using System.Collections.Generic;
using System.Text;

namespace AssignmentMidTask2
```

```
{
    class person
    {
        private string name;
        private string nid;
        private string phoneNumber;
        private int age;
        private BankAccount b;
        internal bool hasAccount = false;



        public person()
        {

        }

        public person(string name, string nid,string phoneNumber,int age)
        {
            this.name = name;
            this.nid = nid;
            this.phoneNumber = phoneNumber;
            this.age = age;
        }

        public string Name
        {
            set
            {
```

```
        if (string.IsNullOrEmpty(value))
        {
            Console.WriteLine("Invalid Input!");
        }
        else
        {
            name = value;
        }

    }
    get
    {
        return name;
    }




}

public string NID
{
    set
    {
        if (string.IsNullOrEmpty(value))
        {
            Console.WriteLine("Invalid Input!");
        }
        else
        {
```

```
            nid = value;
        }


    }
    get
    {
        return nid;
    }

}



public string Number
{
    set
    {
        if (string.IsNullOrEmpty(value))
        {
            Console.WriteLine("Invalid Input!");
        }
        else
        {
            phoneNumber = value;
        }


    }
    get
    {
        return phoneNumber;
```

```
    }



}

public int Age
{
    set
    {
        if (value<=0)
        {
            Console.WriteLine("Invalid Input!");
        }
        else
        {
            age = value;
        }

    }
    get
    {
        return age;
    }



}
```

```
public void showInfo()
{
   Console.WriteLine("Name: {0}",name);
   Console.WriteLine("NID: {0}",nid);
   Console.WriteLine("Phone Number: {0}",phoneNumber);
   Console.WriteLine("Age: {0}",age);
}

public BankAccount createAccount(person userInfo,string accType)
 {
    if (hasAccount == true)
    {
       Console.WriteLine("Already Has an account,Can not create another");

    }
    else
    {

       if (accType.Equals("current"))
       {

          b = new CurrentAccount(userInfo);
          hasAccount = true;

       }
       else if (accType.Equals("savings"))
       {
          b = new SavingsAccount(userInfo);
```

```
            hasAccount = true;

        }
        else if (accType.Equals("overdraft"))
        {
            Console.Write("\n-----Enter an Overdraft amount for {0}: ",Name);
            int amount = Convert.ToInt32(Console.ReadLine());
            b = new OverdraftAccount(userInfo,amount);
            hasAccount = true;
        }


    }

    return b;

 }

public BankAccount getAccount()
{
   return b;
}

public BankAccount changeAccount(BankAccount previousAccount,person userInfo,string accType)
{
   if (hasAccount == false)
   {
      Console.WriteLine("User Does Not have Any Account");
```

```
}
else
{

   if (accType.Equals("current"))
   {

      b = new CurrentAccount(previousAccount,userInfo);



   }
   else if (accType.Equals("savings"))
   {
      b = new SavingsAccount(previousAccount,userInfo);



   }
   else if (accType.Equals("overdraft"))
   {
      Console.Write("Enter an Overdraft amount: ");
      int amount = Convert.ToInt32(Console.ReadLine());
      b = new OverdraftAccount(previousAccount,userInfo,amount);

   }



}
return b;
```

```
        }



    }
}
```

main class

```
using System;

namespace AssignmentMidTask2
{
    class main
    {
        static void Main(string[] args)
        {
            BankAccount b  = new BankAccount();
            BankAccount b1;
            BankAccount b2;
            BankAccount b3;
            BankAccount b4;

            person s1 = new person("Jenny","28928398","01612153345",25);
```

```
b1=s1.createAccount(s1, "savings");
b1=s1.changeAccount(b1,s1,"current");
b1.deposit(1000);

person s2 = new person("Mona", "28928398", "01612153345", 26);
b2 = s2.createAccount(s2, "savings");
b2.deposit(3000);
b2.withdraw(2500);

person s3 = new person("Lisa", "28928398", "01612153345", 23);
b3 = s3.createAccount(s3, "overdraft");
b3.deposit(2000);
b3.withdraw(5000);

person s4 = new person("Anny", "28928398", "01612153345", 23);
b4 = s4.createAccount(s4, "current");
b4.deposit(7000);
b4 =s4.changeAccount(b4,s4, "savings");
b4.withdraw(6500);

b.showAllCustomers();

}
```

```
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Radio
{
    interface RadioPlayer
    {
        void Switch(Boolean on);
        void retune(double frequency);
        void setVolume(int loudness);
        void changeChannel();
    }
    interface MusicPlayer
    {
        void Switch1(Boolean on);
        void play(Boolean on);
        void SetVolume(int loudness);
        void playNext();
        void playPrevious();

    }
```

```
class Music
{
    private string title;
    private string artist;
    private int yearOfRelease;
    private int durationInSec;
    public string Title
    {
        set;
        get;
    }
    public string Artist
    {
        set;
        get;
    }
    public int ReleaseYear
    {
        set;
        get;
    }
    public int Duration
    {
        set;
        get;
    }
    public Music()
    {
        Console.WriteLine("This is emphty COnstructor");
```

```
}
public Music(string title,string artist,int yearOfRelease,int durationInSec)
{
   this.title = title;
   this.artist = artist;
   this.yearOfRelease = yearOfRelease;
   this.durationInSec = durationInSec;
}
void Title1()
{
   Console.WriteLine("Cureent song is :"+this.title);
   Console.WriteLine("Singer Name is :" +this.artist);
   Console.WriteLine("Release year of song is :" +this.yearOfRelease);
   Console.WriteLine("Duration of Song in seconds :" +this.durationInSec);




}
void Title2()
{
   Console.WriteLine("Cureent song is :" + this.title);
   Console.WriteLine("Singer Name is :" + this.artist);
   Console.WriteLine("Release year of song is :" + this.yearOfRelease);
   Console.WriteLine("Duration of Song in seconds :" + this.durationInSec);




}
class play:RadioPlayer
```

```
{
    public void Switch(Boolean on)
    {
        if (on == true)
        {
            Console.WriteLine(" Radio Statrted");
        }
        else
        {
            Console.WriteLine("Stopped");
        }
    }

    public void retune(double frequency)
    {
        Console.WriteLine(" frequency is :" +frequency);
    }
    public void setVolume(int loudness)
    {
        Console.WriteLine("Volume is :", +loudness);
    }
    public void changeChannel()
    {
        Console.WriteLine("The Channel is changed");
    }
    public void Switch1(Boolean on)
    {
        if (on == true)
        {
```

```
        Console.WriteLine(" Music Statrted");
    }
    else
    {
        Console.WriteLine("Stopped");
    }
}
public void Play(Boolean on)
{
    if (on == true)
    {
        Console.WriteLine("Play Music ");
    }
    else
    {
        Console.WriteLine("Stopped");
    }
}
public void SetVolume(int loudess)
{
    Console.WriteLine("The Volume of the music is " + loudess);
}
public void playNext()
{
    Console.WriteLine("Next Song is Kabira");
}
public void playPrevious()
{
    Console.WriteLine("Previous Song is Mehboob");
```

```
    }
}
    class pogram
{
    static void Main(String []args)
    {

        Music c1 = new Music();
        c1.Title = "PAGALPANTI";
        c1.Artist = "BENNY";
        c1.ReleaseYear = 2012;
        c1.Duration = 500;
        Console.WriteLine("New Title:" + c1.Title);
        Console.WriteLine("Artist:" + c1.Artist);
        Console.WriteLine("RELEASE YEAR:" + c1.ReleaseYear);
        Console.WriteLine("Duration:" + c1.Duration);



        Music c2 = new Music("kabira", "Arjit", 2013, 600);
        c2.Title2();
        Music c3 = new Music("MEHBOOB", "Atif", 2015, 400);
        c3.Title1();
        Console.WriteLine("Radio Status");
        play s1 = new play();
        s1.Switch(true);
        s1.setVolume(50);
        s1.changeChannel();
        s1.Switch1(true);
```

```
            s1.Play(true);
            s1.SetVolume(100);
            s1.playNext();
            s1.playPrevious();



        }
      }

   }
}
```