Day 5 – C# - 04.08.2023

1) Practice Program
ElectricReading.cs

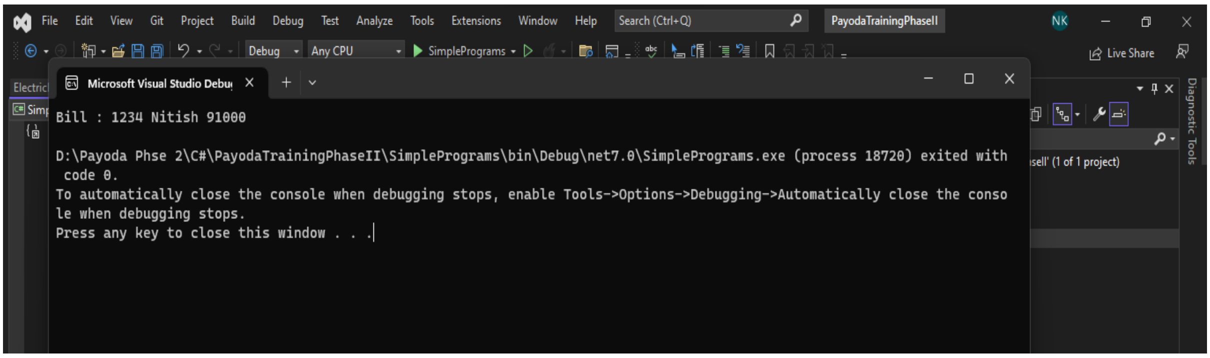
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Xml;

namespace SimplePrograms
{
    internal class ElectricReading
    {
        private int consumernumber,previousreading,currentreading;
        private string consumername,consumertype;

        public ElectricReading(int consumernumber, string consumername, int
currentreading, int previousreading, string consumertype)
        {
            Consumernumber = consumernumber;
            Previousreading = previousreading;
            Currentreading = currentreading;
            Consumername = consumername;
            Consumertype = consumertype;
        }

        public int Consumernumber { get => consumernumber; set =>
consumernumber = value; }
        public int Previousreading { get => previousreading; set =>
previousreading = value; }
        public int Currentreading { get => currentreading; set => currentreading =
value; }
        public string Consumername { get => consumername; set =>
consumername = value; }
        public string Consumertype { get => consumertype; set => consumertype
= value; }

        public int CalculateBill()
        {
            int consumption = Currentreading - Previousreading;
    int billamt = 0;
    if (Consumertype.Equals("Domestic"))
    {

        if (consumption <= 100)
        {
            billamt = 0;
        }
        else if (consumption>100 && consumption<=200){
            billamt = (consumption - 100) * 2;
        }
        else if (consumption > 200 && consumption <= 500)
        {
            billamt = (consumption - 100) * 5;
        }
        else if (consumption > 500)
        {
            billamt = (consumption - 100) * 10;
        }


    }
    else if (Consumertype.Equals("Commercial"))
    {
        if (consumption <= 100)
        {
            billamt = 10;
        }
        else if (consumption > 100 && consumption <= 200)
        {
            billamt = (consumption - 100) * 20;
        }
        else if (consumption > 200 && consumption <= 500)
        {
            billamt = (consumption - 100) * 50;
        }
        else if (consumption > 500)
        {
            billamt = (consumption - 100) * 100;
        }
    }
    return billamt;
}
```

```
        /* public void DisplayBillDetails()
         {
             int billamt=CalculateBill();
             Console.WriteLine("Bill : " + consumernumber + " " + consumername +
" " + billamt);
         }*/

    }
}
```

Program.cs

```
using SimplePrograms;

/*int consumernumber = Convert.ToInt32(Console.ReadLine());
string? consumername = Console.ReadLine();
int currentreading = Convert.ToInt32(Console.ReadLine());
int previousreading = Convert.ToInt32(Console.ReadLine());
string? consumertype = Console.ReadLine();*/

ElectricReading electricReading = new ElectricReading(1234, "Nitish", 10000,
800, "Domestic");
int billamt = electricReading.CalculateBill();
Console.WriteLine($"Bill : " + $"{electricReading.Consumernumber}
{electricReading.Consumername}  {billamt}");
```

Output



```
Bill : 1234 Nitish 91000

D:\Payoda Phse 2\C#\PayodaTrainingPhaseII\SimplePrograms\bin\Debug\net7.0\SimplePrograms.exe (process 18720) exited with
 code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

2)Assignment 1

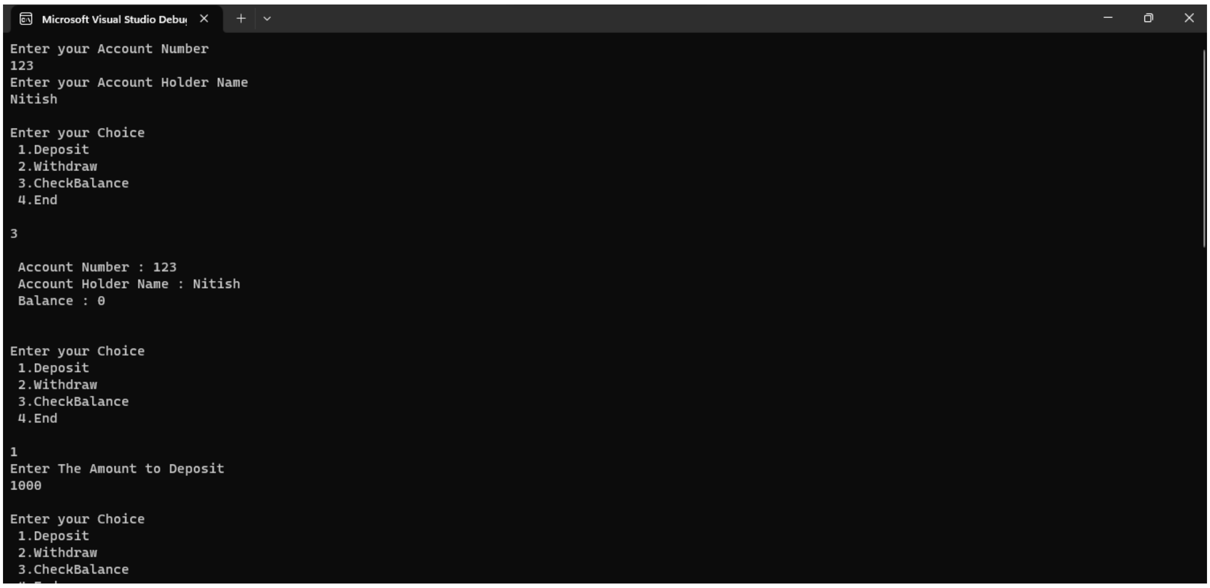
Program.cs

```
using Bank;
using System.Xml.Serialization;

Console.WriteLine("Enter your Account Number");
int AccountNumber = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter your Account Holder Name");
string? AccountHolderName = Console.ReadLine();
BankAccount Account = new
BankAccount(AccountNumber,AccountHolderName);
int choice;
do
{
    Console.WriteLine("\nEnter your Choice \n 1.Deposit \n 2.Withdraw \n
3.CheckBalance \n 4.End\n");
    choice = Convert.ToInt32(Console.ReadLine());
    switch (choice)
    {
        case 1:
            Account.deposit();
            break;
        case 2:
            Account.withdraw();
            break;
        case 3:
            Account.PrintBalance();
            break;
        case 4:
            break;
        default:
            Console.WriteLine("\nYour choice is wrong\nPlease try again...\n");
            break;
    }
}while(choice!=4);
```

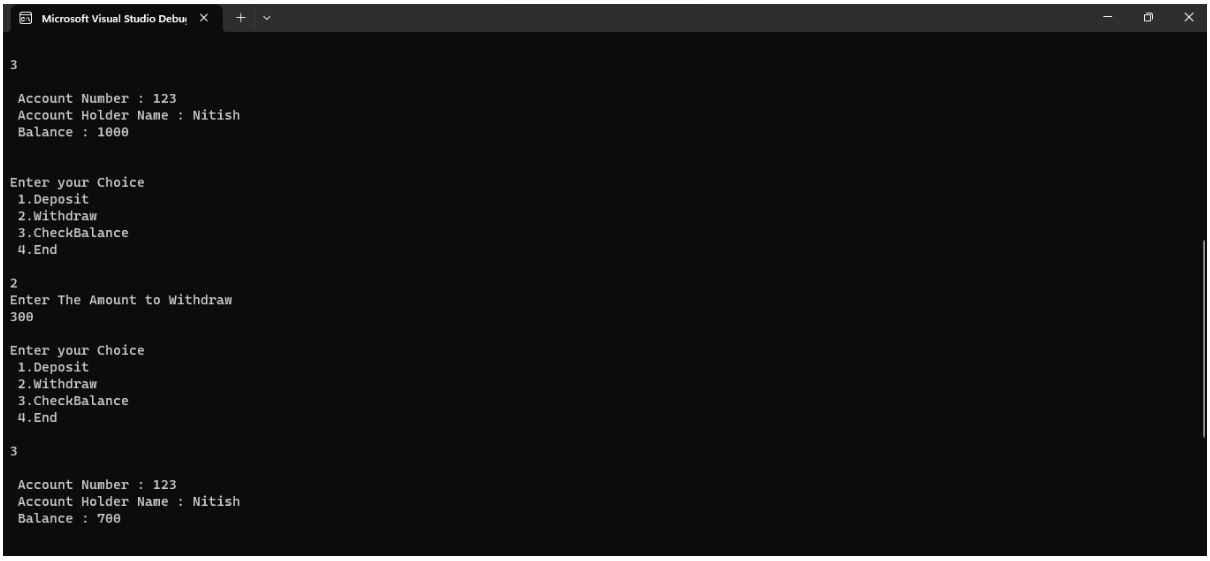
BankAccount.cs

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

namespace Bank
{
    internal class BankAccount
    {
        private readonly int AccountNumber;
        private string AccountHolderName;
        public static int Balance=0;
        public BankAccount(int AccountNumber,string AccountHolderName)
        {
            this.AccountNumber = AccountNumber;
            AccountHolderName1 = AccountHolderName;
        }
        public string AccountHolderName1 { get => AccountHolderName; set =>
AccountHolderName = value; }

        public void deposit()
        {
            Console.WriteLine("Enter The Amount to Deposit");
            int Dep_Amount = Convert.ToInt32(Console.ReadLine());
            Balance = Balance + Dep_Amount;
        }

        public void withdraw()
        {
            Console.WriteLine("Enter The Amount to Withdraw");
            int Wit_Amount = Convert.ToInt32(Console.ReadLine());
            if (Wit_Amount <= Balance)
            {
                Balance = Balance - Wit_Amount;
            }
            else
            {
                Console.WriteLine("\n Insufficient Balance \n");
            }
        }

        public void PrintBalance()
        {
            Console.WriteLine($"\n Account Number : {AccountNumber} \n
Account Holder Name : {AccountHolderName1} \n Balance : {Balance} \n");
        }
    }
}
```

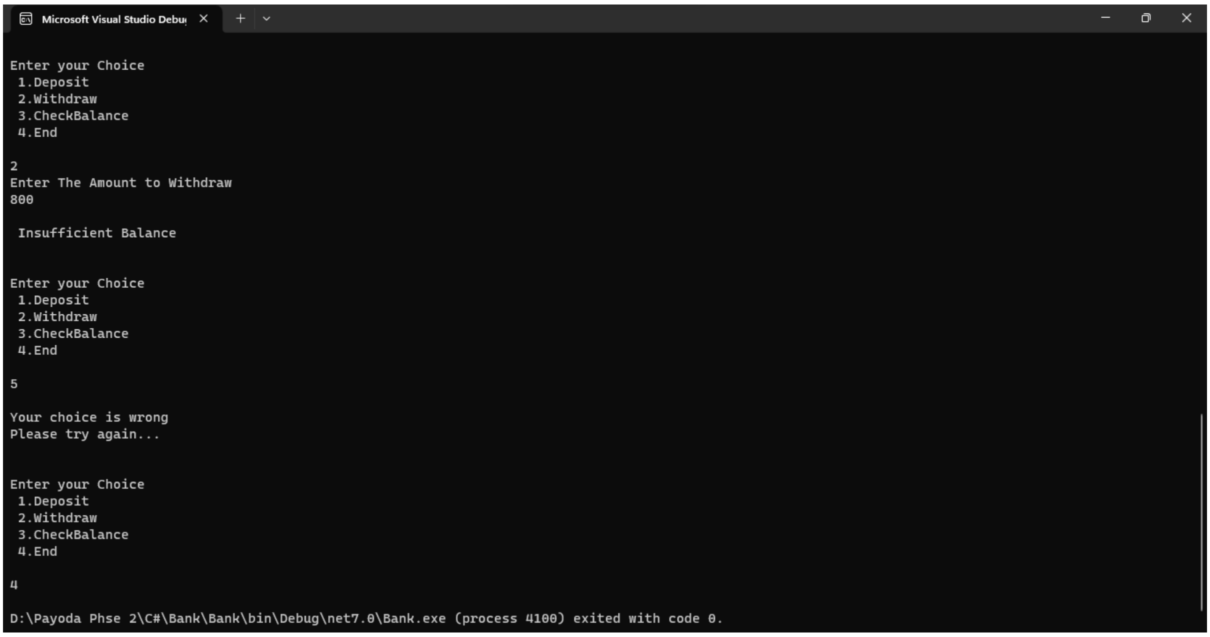
Output



```
Enter your Account Number
123
Enter your Account Holder Name
Nitish

Enter your Choice
 1.Deposit
 2.Withdraw
 3.CheckBalance
 4.End

3

 Account Number : 123
 Account Holder Name : Nitish
 Balance : 0


Enter your Choice
 1.Deposit
 2.Withdraw
 3.CheckBalance
 4.End

1
Enter The Amount to Deposit
1000

Enter your Choice
 1.Deposit
 2.Withdraw
 3.CheckBalance
```



```
3

 Account Number : 123
 Account Holder Name : Nitish
 Balance : 1000


Enter your Choice
 1.Deposit
 2.Withdraw
 3.CheckBalance
 4.End

2
Enter The Amount to Withdraw
300

Enter your Choice
 1.Deposit
 2.Withdraw
 3.CheckBalance
 4.End

3

 Account Number : 123
 Account Holder Name : Nitish
 Balance : 700
```



```
Enter your Choice
 1.Deposit
 2.Withdraw
 3.CheckBalance
 4.End

2
Enter The Amount to Withdraw
800

 Insufficient Balance


Enter your Choice
 1.Deposit
 2.Withdraw
 3.CheckBalance
 4.End

5

Your choice is wrong
Please try again...


Enter your Choice
 1.Deposit
 2.Withdraw
 3.CheckBalance
 4.End

4

D:\Payoda Phse 2\C#\Bank\Bank\bin\Debug\net7.0\Bank.exe (process 4100) exited with code 0.
```

3)Assignment 2

Program.cs

```
using Library;
using System.Collections;


Book book1 = new Book(1, "PS", "Kalki", true);
Book book2 = new Book(2, "Twilight", "Stephnie Mayor", true);
Book book3 = new Book(3, "Harry Potter", "JK Rowling", false);

List<Book> Books = new List<Book>();
Books.Add(book1);
Books.Add(book2);
Books.Add(book3);
LibraryClass lib=new LibraryClass(Books);
lib.ReturnBook(book3);
lib.BorrowBook(book1);
lib.PrintBook();
```

Book.cs

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

namespace Library
{
    internal class Book
    {
        private readonly int bookId;
        private string Title;
        private string Author;
        private bool IsAvailable;

        public Book(int bookId, string title, string author,bool IsAvailable)
        {
            this.bookId = bookId;
            Title1 = title;
            Author1 = author;
            IsAvailable1 = IsAvailable;
        }
        public string Title1 { get => Title; set => Title = value; }
        public string Author1 { get => Author; set => Author = value; }
        public bool IsAvailable1 { get => IsAvailable; set => IsAvailable = value;
}



    }
}
```

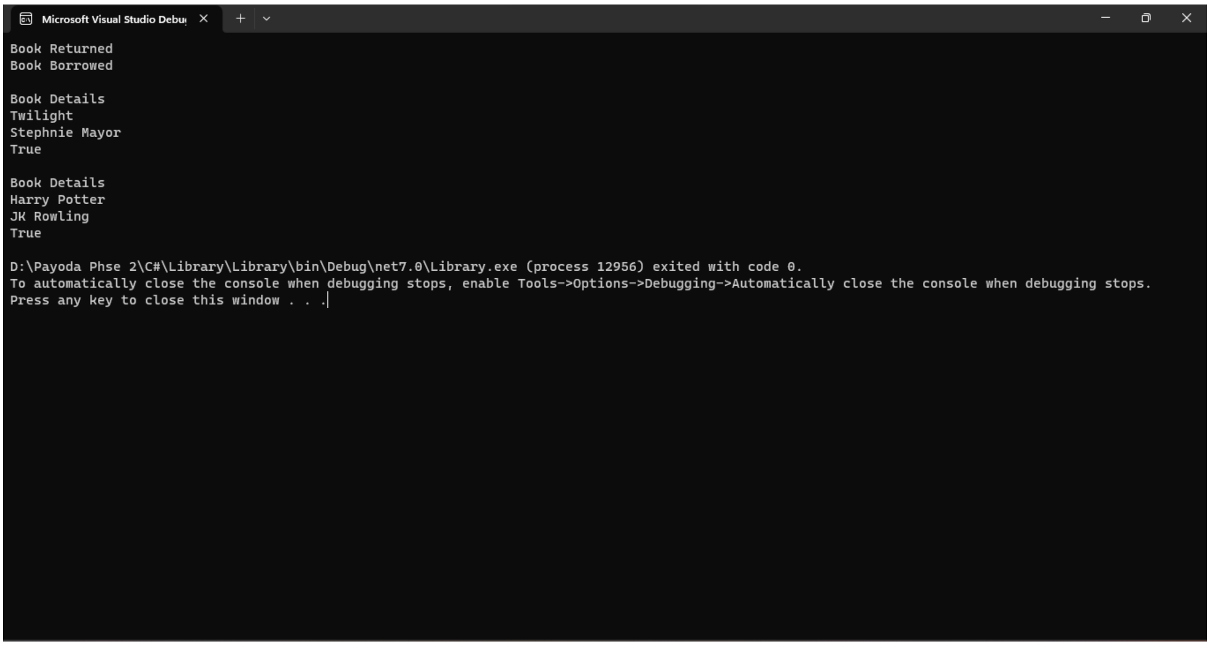
LibraryClass.cs

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

namespace Library
{
    internal class LibraryClass
```

```
{
    private readonly List<Book> book;
    public LibraryClass(List<Book> book)
    {
        this.book = book;

    }

    internal List<Book> Book => book;

    public void BorrowBook(Book book)
    {
        if (book.IsAvailable1 == true)
        {
            Console.WriteLine("Book Borrowed");
            book.IsAvailable1 = false;
        }
        else
        {
            Console.WriteLine("Book is not Available");
        }
    }

    public void ReturnBook(Book book)
    {
        if (book.IsAvailable1 == false)
        {
            Console.WriteLine("Book Returned");
            book.IsAvailable1 = true;
        }
        else
        {
            Console.WriteLine("Book is not Available");
        }
    }

    public void PrintBook()
    {
        foreach (Book book in book)
        {
            if (book.IsAvailable1 == true)
            {
                Console.WriteLine();
```

```
                    Console.WriteLine("Book Details");
                    Console.WriteLine(book.Title1);
                    Console.WriteLine(book.Author1);
                    Console.WriteLine(book.IsAvailable1);
                }
            }
        }
    }
}
```

Output:



```
Book Returned
Book Borrowed

Book Details
Twilight
Stephnie Mayor
True

Book Details
Harry Potter
JK Rowling
True

D:\Payoda Phse 2\C#\Library\Library\bin\Debug\net7.0\Library.exe (process 12956) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```