WeBankApp Assignment Solution: **1**
Customer.cs 1
SavingsAccountHolder.cs 2
CurrentAccountHolder.cs 3
Banker.cs: 4
Bank.cs 6
Transaction.cs 7
Program.cs 9

WeBankApp Assignment Solution:

Customer.cs

```
namespace WeBankBusinessLayer
{
    public class Customer
    {
        public int CustomerId { get; set; }
        public string Name { get; set; }
        public int Age { get; set; }
        public string Gender { get; set; }
        public string PhoneNumber { get; set; }

        public Customer(int customerId, string name, int age, string
gender,
            string phoneNumber)
        {
            CustomerId = customerId;
            Name = name;
            Age = age;
            Gender = gender;
```

```
            PhoneNumber = phoneNumber;
        }

        public virtual string DisplayCustomerInfo()
        {
            return CustomerId + " " + Name + " " + Age + " " +
PhoneNumber + " " + Gender;
        }
    }
}
```

SavingsAccountHolder.cs

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

namespace WeBankBusinessLayer
{
    public class SavingsAccountHolder : Customer
    {
        public double InterestRate { get; set; }
        public double Balance { get; set; }

        public SavingsAccountHolder(double interestRate, double
balance,
            int customerId, string name, int age, string gender,
            string phoneNumber) : base(customerId, name, age, gender,
phoneNumber)
        {
            InterestRate = interestRate;
            Balance = balance;
```

```
        }

        public override string DisplayCustomerInfo()
        {
            string info = base.DisplayCustomerInfo() + " " + InterestRate
+ " " + Balance;
            return info;
        }
    }
}
```

CurrentAccountHolder.cs

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

namespace WeBankBusinessLayer
{
    public class CurrentAccountHolder : Customer
    {
        public double OverdraftLimit { get; set; }

        public CurrentAccountHolder(double overdraftLimit,
            int customerId, string name, int age, string gender,
            string phoneNumber) : base(customerId, name, age, gender,
phoneNumber)
        {
            OverdraftLimit = overdraftLimit;
        }
```

```
        public override string DisplayCustomerInfo()
        {
            string info = base.DisplayCustomerInfo() + " " +
OverdraftLimit;
            return info;
        }
    }
}
```

Banker.cs:

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

namespace WeBankBusinessLayer
{
    public class Banker
    {
        public int BankerId { get; set; }
        public string Name { get; set; }
        public string Branch { get; set; }
        public string PhoneNumber { get; set; }

        public Banker(int bankerId, string name, string branch, string
phoneNumber)
        {
            BankerId = bankerId;
            Name = name;
            Branch = branch;
            PhoneNumber = phoneNumber;
        }
```

```
        public bool UpdateBankerDetails(string newBranch)
        {
            if (newBranch != Branch)
            {
                Branch = newBranch;
                return true;
            }
            return false;
        }

        public bool UpdateBankerDetails(string newBranch, string
newPhoneNumber)
        {
            bool status = false;
            if (newBranch != Branch)
            {
                Branch = newBranch;
                status = true;
            }
            if (newPhoneNumber != PhoneNumber)
            {
                PhoneNumber = newPhoneNumber;
                status = true;
            }
            return status;
        }

        public string DisplayBankerInfo()
        {
            return BankerId + " " + Name + " " + Branch + " " +
PhoneNumber;
        }
    }
}
```

Bank.cs

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

namespace WeBankBusinessLayer
{
   public class Bank
   {
      public string BankId { get; set; }
      public string Name { get; set; }
      public string Location { get; set; }
      public Banker Banker { get; set; }

      private static int counter;

      static Bank()
      {
         counter = 1000;
      }

      public Bank(string name, string location, Banker banker)
      {
         BankId = "B" + ++counter;
         Name = name;
         Location = location;
         Banker = banker;
      }

      public string DisplayBankInfo()
      {
         return BankId + " " + Name + " " + Location + " " +
Banker.Name;
```

```
        }
    }
}
```

Transaction.cs

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

namespace WeBankBusinessLayer
{
    public class Transaction
    {
        public string TransactionId { get; set; }
        public DateTime TransactionDate { get; set; }
        public double Amount { get; set; }
        public string Type { get; set; }
        public string Status { get; set; }

        private static int counter;

        public Transaction()
        {

        }
        static Transaction()
        {
            counter = 500;
        }

        public string DisplayTransactionDetails()
```

```
        {
            return TransactionId + " " + Type + " " + Amount + " " +
Status + " " + TransactionDate;
        }

        public string ProcessTransaction(SavingsAccountHolder customer,
Banker banker, double amount,
            string type)
        {
            try
            {
                if (customer != null && banker != null)
                {
                    if(type == "Debit")
                    {
                        if(customer.Balance >= amount)
                        {
                            TransactionId = "D" + ++counter;
                            Amount = amount;
                            Type = type;
                            TransactionDate = DateTime.Now;
                            Status = "Completed";
                        }
                    }
                    else if(type == "Credit")
                    {
                        TransactionId = "C" + ++counter;
                        Amount = amount;
                        Type = type;
                        TransactionDate = DateTime.Now;
                        Status = "Completed";
                    }

                    return "Transaction completed for customer " +
customer.Name
```

```
                    + " with banker " + banker.Name + " with Transaction
ID : " +
                    TransactionId;
            }
            return "Transaction could not be completed";
        }
        catch (Exception ex)
        {
            return "Some error occurred, transaction failed!";
        }

    }
  }
}
```

Program.cs

```
using WeBankBusinessLayer;

namespace WeBankConsoleApp
{
  internal class Program
  {
    static void Main(string[] args)
    {
      SavingsAccountHolder savingsAccountHolder = new
SavingsAccountHolder(2.5,
        50000, 101, "Anna Miller", 34, "Female", "9999999999");

      CurrentAccountHolder currentAccountHolder = new
CurrentAccountHolder(20000,
        102, "Frank Lawson", 29, "Male", "8888888888");
```

```
        Banker banker = new Banker(1, "Katie Otto", "Church Street",
"7777777777");

        Bank bank = new Bank("We Trust Bank", "New York", banker);

        Console.WriteLine("Savings Account Holder details : " +
savingsAccountHolder.DisplayCustomerInfo());


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

        Console.WriteLine("Current Account Holder details : " +
currentAccountHolder.DisplayCustomerInfo());


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

        Console.WriteLine("Banker Details : " +
banker.DisplayBankerInfo());

        Console.WriteLine("Updating Banker Contact Info");

        banker.UpdateBankerDetails("Mall Road");

        Console.WriteLine("Banker Details : " +
banker.DisplayBankerInfo());


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

        Console.WriteLine("Bank Details: " +
bank.DisplayBankInfo());
```

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

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

            Transaction transactionOne = new Transaction();
            Transaction transactionTwo = new Transaction();

            string messageOne =
transactionOne.ProcessTransaction(savingsAccountHolder, banker,
1000, "Debit");
            Console.WriteLine("Transaction One");
            Console.WriteLine(messageOne);


Console.WriteLine("------------------------------------------------------
");
            Console.WriteLine("Transaction Two");
            string messageTwo =
transactionTwo.ProcessTransaction(savingsAccountHolder, banker,
3000, "Credit");
            Console.WriteLine(messageTwo);
        }
    }
}
```