WIN\_DAY\_1

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace win\_day1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void btn\_addcustomer\_Click(object sender, EventArgs e)

{

MessageBox.Show("hello from dot net ");

}

}

}

CONSOLE \_DAY\_1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace console\_day1

{

class Program

{

static void Main(string[] args)

{

/\* Console.WriteLine("enter your name :");

string name;

name = Console.ReadLine();

Console.WriteLine("your name is :" + name);

Console.ReadLine();\*/

Console.WriteLine(DateTime.Today);

Console.WriteLine(DateTime.Now);

Console.WriteLine(DateTime.DaysInMonth(28,2017));

Console.ReadLine();

}

}

}

CONSOLE\_CSHARP

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_csharp

{

class Program

{

static void Main(string[] args)

{

/\* string employeename;

int employeesalary=0;

string employeecity;

DateTime employeedoj;

int employeeexp = 0;

Console.WriteLine("enter employee name");

employeename = Console.ReadLine();

Console.WriteLine("enter salary");

employeesalary = Convert.ToInt32( Console.ReadLine());

Console.WriteLine("enter employee city");

employeecity = Console.ReadLine();

Console.WriteLine("enter doj"); //mm/dd/yyy

employeedoj=Convert.ToDateTime(Console.ReadLine());

TimeSpan t = DateTime.Now - employeedoj;

employeeexp = Convert.ToInt32(t.TotalDays / 365);

Console.WriteLine("name :" + employeename);

Console.WriteLine("salary:" + employeesalary);

Console.WriteLine("city :" + employeecity);

Console.WriteLine(DateTime.Now );

Console.WriteLine("t :" + t);

Console.WriteLine("doj :" + employeedoj.ToShortDateString());

Console.WriteLine("exp:" + employeeexp); \*/

int count = 50;

if (count > 100)

{

Console.WriteLine(">100");

}

else

{

Console.WriteLine(false);

}

int i = 0;

while (i < 10)

{

Console.WriteLine(i);

i++;

}

for (int c = 0; c < 10; c++)

{

Console.WriteLine(c);

}

int option=1;

switch (option)

{

case 1:

Console.WriteLine("case 1");

break;

case 2:

Console.WriteLine("case 2");

break;

default:

Console.WriteLine("default case");

break;

}

int[] marks = new int[5];

marks[0] = 23;

marks[1] = 45;

marks[2] = 29;

marks[3] = 90;

marks[4] = 100;

int[] salaries = { 1, 2, 3 };

for (int j = 0; j < marks.Length; j++)

{

Console.WriteLine(marks[j]);

}

Console.ReadLine();

}

}

}

CONSOLE\_CSHARPBUILTFUNC

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace console\_csharp\_builtinfun

{

class Program

{

static void Main(string[] args)

{

/\*char ch = 'A';

int num = Convert.ToInt32(ch);

Console.WriteLine(num);

num+=32;

Console.WriteLine(Convert.ToChar(num)); \*/

string Str1 = "abcd";

Console.WriteLine(Str1.ToUpper());

/\* string str = "Hello world";

int Cu=0;

int Cl=0;

foreach (char c in str)

{

if (char.IsUpper(c))

{

Cu++;

}

else if (char.IsLower(c))

{

Cl++;

}

Console.WriteLine(c);

}

Console.WriteLine(Cu );

Console.WriteLine(Cl); \*/

/\* string str = "89 61 62 90 64";

string[] str\_array =str.Split(' ');

int[] marks =Array.ConvertAll(str\_array,int.Parse);

Array.Sort(marks);

Array.Reverse(marks);

foreach (int m in marks)

{

Console.WriteLine(m);

} \*/

Console.ReadLine();

}

}

}

CONSOLE\_BANKSYSTEM

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace console\_bank\_system

{

class Program

{

static void Main(string[] args)

{

/\* int x = Console.Read();

Console.WriteLine(x); \*/

int accountid;

string customername;

string customeraddress;

DateTime customerdob;

string accounttype;

double accountbalance;

Console.WriteLine("enter the Account id");

accountid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter Name: ");

customername=Console.ReadLine();

customername=customername.ToUpper();

Console.WriteLine("Enter Address: ");

customeraddress = Console.ReadLine();

Console.WriteLine("Enter DOB(mm/dd/yyyy: ");

customerdob = Convert.ToDateTime(Console.ReadLine());

Console.WriteLine("Enter type: ");

accounttype = Console.ReadLine();

Console.WriteLine("Enter AccountBalance: ");

accountbalance = Convert.ToDouble(Console.ReadLine());

int option;

do

{

Console.WriteLine("enter an option: 1. deposit 2.withdrawal 3. check balance 4. customer detail 5. exit ");

option = Convert.ToInt32(Console.ReadLine());

switch (option)

{

case 1:

Console.WriteLine("enter the amount to be deposited");

int depamt = Convert.ToInt32(Console.ReadLine());

if (depamt < 500)

{

Console.WriteLine("Please deposit a amount greater than 500");

}

else

{

accountbalance = accountbalance + depamt;

}

break;

case 2:

Console.WriteLine("enter the amount to be withdrawn");

int withdrawnamt = Convert.ToInt32(Console.ReadLine());

if (withdrawnamt > 5000 || accountbalance < withdrawnamt)

{

Console.WriteLine("Amount cannot be withdrawn");

}

else

{

accountbalance = accountbalance - withdrawnamt;

}

break;

case 3:

Console.WriteLine("Account Balance : " + accountbalance);

break;

case 4:

Console.WriteLine("Customer Details :");

Console.WriteLine("Customer Name :" + customername);

Console.WriteLine("Customer Address :" + customeraddress);

TimeSpan t = DateTime.Now - customerdob;

int age = Convert.ToInt32(t.TotalDays / 365);

Console.WriteLine("Customer Age :" + age);

break;

case 5:

Console.WriteLine("Exit");

break;

}

} while (option != 5);

Console.ReadLine();

}

}

}

PRACTICE \_ DAY1

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace practice

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void sum\_Click(object sender, EventArgs e)

{

DateTime t = DateTime.Today;

textBox2.Text = Convert.ToString((Convert.ToInt32(t-(Convert.ToDateTime(textBox1.Text))))/365);

}

}

}

CONSOLE\_JAGGEDARRAY

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_jaggedarray

{

class Program

{

static void Main(string[] args)

{

int[][] jaggedarray = new int[3][];

jaggedarray[0] = new int[2];

jaggedarray[1] = new int[3];

jaggedarray[2] = new int[1];

jaggedarray[0][0] = 25;

jaggedarray[0][1] = 30;

jaggedarray[1][0] = 12;

jaggedarray[1][1] = 31;

jaggedarray[1][2] = 55;

jaggedarray[2][0] = 322;

foreach (int[] array in jaggedarray)

{

foreach (int marks in array)

{

Console.WriteLine(marks);

}

}

Console.ReadLine();

}

}

}

CONSOLE\_OOP1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("enter the customer details:");

Console.WriteLine("enter the customer first name :");

string customerFname = Console.ReadLine();

Console.WriteLine("enter the customer last name :");

string customerLname = Console.ReadLine();

Console.WriteLine("enter the customer Account id :");

int accountid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("enter the customer Id:");

int customerid= Convert.ToInt32(Console.ReadLine());

Console.WriteLine("enter the customer Account balance:");

int accountbalance = Convert.ToInt32(Console.ReadLine());

customer obj = new customer(customerid,customerFname,customerLname,accountbalance,accountid);

string Name=obj.customerFullname();

Console.WriteLine("Customer Nmae :" + Name);

int Balance = obj.balance();

Console.WriteLine("Account Balance :" + Balance);

Console.WriteLine("enter 1 to update or 0 to exit :");

int option = Convert.ToInt32(Console.ReadLine());

if (option == 1)

{

Console.WriteLine("enter the customer id to update the name");

int id = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Customer first Nmae :");

customerFname = Console.ReadLine();

Console.WriteLine("Customer Last Nmae :");

customerLname = Console.ReadLine();

bool test = obj.updatename(id, customerFname, customerLname);

if (test)

{

Console.WriteLine("updated");

Console.WriteLine("thank u");

}

else

{

Console.WriteLine("Not updated");

Console.WriteLine("thank u");

}

}

else

{

Console.WriteLine("thank u");

}

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop1

{

class customer

{

private int customerid;

private string customerFname;

private string customerLname;

private int accountid;

private int accountbalance;

public customer(int customerid,string customerFname,

string customerLname,int accountbalance,int accountid):this(customerid)

{

Console.WriteLine("constructor called");

this. customerid = customerid;

this. customerFname = customerFname;

this. customerLname = customerLname;

this. accountbalance = accountbalance;

this .accountid = accountid;

}

public customer(int customerid)

{

this.customerid = customerid;

Console.WriteLine("existing ");

}

public string customerFullname()

{

return customerFname + " " + customerLname;

}

public int balance()

{

return accountbalance;

}

public bool updatename(int customerid, string customerFname, string customerLname)

{

if (this.customerid == customerid)

{

this.customerFname = customerFname;

this.customerLname = customerLname;

return true;

}

else

{

return false;

}

}

}

}

CONSOLE\_OOP2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop2

{

class Program

{

static void Main(string[] args)

{

Student obj = new Student(100, "vifhh", 60);

Console.WriteLine(obj.Pstudentid);

Console.WriteLine(obj.Pstudentname);

Console.WriteLine(obj.Pstudentmarks);

Console.WriteLine(obj.Pstudentstatus);

obj.Pstudentname = Console.ReadLine() ;

obj.Pstudentmarks = 1000;

Console.WriteLine(obj.Pstudentmarks);

Console.WriteLine(obj.Pstudentstatus);

Console.WriteLine(obj.Pstudentname);

obj.Pstudentmarks = 75;

Console.WriteLine(obj.Pstudentmarks);

Console.WriteLine(obj.Pstudentstatus);

/\* customer c = new customer();

c.getdata();

Test obj = new Test(1000,"abc");

Customernameupdatestatus status = obj.update(100, "");

if (status==Customernameupdatestatus.Updated)

{

Console.WriteLine("yes updated");

}

else if(status==Customernameupdatestatus.invalidname)

{

Console.WriteLine("Customer name is invalid");

}

else if(status==Customernameupdatestatus.Invalidcustomerid)

{

Console.WriteLine(" customer id is invalid");

}

int salary = obj.getsalary(10000, 31); //normal way

Console.WriteLine(salary);

salary = obj.getsalary(10000); //oneoptional parameter

Console.WriteLine(salary);

salary = obj.getsalary(no:31,perdayssalary: 10000); //named parameter

Console.WriteLine(salary);\*/

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop2

{

class Student

{

private int studentid;

private string studentname;

private int studentmarks;

private bool studentstatus;

public int Pstudentid

{

get

{

return studentid;

}

}

public string Pstudentname

{

get

{

return studentname;

}

set

{

studentname = value;

}

}

public int Pstudentmarks

{

get

{

return studentmarks;

}

set

{

if (value > 100 || value < 0)

{

studentmarks = 0;

}

else

{

studentmarks = value;

}

if (studentmarks >= 50)

{

studentstatus = true;

}

else

{

studentstatus = false;

}

}

}

public bool Pstudentstatus

{

get

{

return studentstatus;

}

}

public Student(int studentid, string studentname, int studentmarks)

{

this.studentid = studentid;

this.studentname = studentname;

this.studentmarks = studentmarks;

if (this.studentmarks >= 50)

{

this.studentstatus = true;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop2

{

partial class customer

{

public void updatedata()

{

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop2

{

partial class customer

{

public void getdata()

{

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop2

{

enum Customernameupdatestatus

{

Updated,Invalidcustomerid,invalidname

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop2

{

class Test

{

private string customername;

private int customerid;

public Test(int customerid, string customername)

{

this.customerid = customerid;

this.customername = customername;

}

public Customernameupdatestatus update(int customerid, string customernewname)

{

if (this.customerid == customerid)

{

if (customernewname != "")

{

this.customername = customernewname;

return Customernameupdatestatus.Updated;

}

else

{

return Customernameupdatestatus.invalidname;

}

}

else

{

return Customernameupdatestatus.Invalidcustomerid;

}

}

public int getsalary(int perdayssalary, int no = 30)

{

int total = perdayssalary \* no;

return total;

}

}

}

ASSIGNMENT\_DAY2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assignment\_2

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the customer details:");

Console.WriteLine("Enter the customer id:");

int account\_id=Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the customer Name:");

string customer\_name = Console.ReadLine();

Console.WriteLine("Enter the customer dob (mm/dd/yyyy):");

DateTime customer\_dob =Convert.ToDateTime(Console.ReadLine());

Account obj = new Account(account\_id,customer\_name,customer\_dob);

int option;

do

{

Console.WriteLine("Enter the option 1.getbalance 2.withdraw 3.deposit 4.customer details: 5.exit");

option = Convert.ToInt32(Console.ReadLine());

switch (option)

{

case 1:

double balance = obj.getbalance();

Console.WriteLine(" Available Balance:" + balance);

break;

case 2:

Console.WriteLine("Enter the amount to be withdrawn");

int amountw = Convert.ToInt32(Console.ReadLine());

bool resultw = obj.withdraw(amountw);

if (resultw)

{

Console.WriteLine(" Amount have been successfully withdrawn");

}

else

{

Console.WriteLine("Enter the correct amount to be withdrawn:");

}

break;

case 3:

Console.WriteLine("Enter the amount to be deposited");

int amountd = Convert.ToInt32(Console.ReadLine());

bool resultd = obj.deposit(amountd);

if (resultd)

{

Console.WriteLine(" Amount have been successfully deposited");

}

else

{

Console.WriteLine("Enter the correct amount to be deposit:");

}

break;

case 4:

Console.WriteLine(" Customer details:");

Console.WriteLine("Account id:" + obj.paccount\_id);

Console.WriteLine("Customer name:" + obj.pcustomer\_name);

Console.WriteLine("Customer DOB:" + obj.pcustomer\_dob);

Console.WriteLine("Account Balance:" + obj.paccount\_balance);

break;

case 5:

Console.WriteLine("THANK YOU");

break;

}

} while (option != 5);

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assignment\_2

{

class Account

{

private int account\_id;

private string customer\_name;

private DateTime customer\_dob;

private double account\_balance;

public Account(int account\_id,string customer\_name,DateTime customer\_dob)

{

this.account\_id=account\_id;

this.customer\_name=customer\_name;

this.customer\_dob=customer\_dob;

}

public int paccount\_id

{

get

{

return account\_id;

}

}

public string pcustomer\_name

{

get

{

return customer\_name;

}

}

public DateTime pcustomer\_dob

{

get

{

return customer\_dob;

}

}

public double paccount\_balance

{

get

{

return account\_balance;

}

}

public double getbalance()

{

return account\_balance;

}

public bool withdraw(int amountw)

{

if (amountw > account\_balance)

{

return false;

}

else

{

account\_balance = account\_balance - amountw;

return true;

}

}

public bool deposit(int amountd)

{

if (amountd > 0)

{

account\_balance = account\_balance + amountd;

return true;

}

else

{

return false;

}

}

}

}

CONSOLE\_OOP 3

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop3

{

class Program

{

static employee empobj = new employee(); //static is empobj variable (an instance) size foe emp obj os used to store pointer is 2 bytes //high frequrncy heap

static void Main(string[] args)

{

project p = new project();

employee.count = 10;

Console.WriteLine(employee.count);

employee obj = new employee();

obj.Employee\_id = 1001;

obj.update();

obj.update();

Console.WriteLine(employee.count);

employee.count = 100;

Console.WriteLine(employee.count);

int i= employee.getdata();

Console.WriteLine(i);

obj.update();

obj.update();

obj.setproject(p);

employee obj2 = new employee();

p = null; // its not a memory leakage

obj = null; //memory leakage

obj2 = null;//memory leakage

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop3

{

class employee

{

public int Employee\_id;

public static int count;

public project empproject;

static employee()

{

count = 5;

}

public void setproject(project p)

{

empproject = p;

}

public void update()

{

employee.count++;

}

public static int getdata() // in this function we cannot use employee id directly

{

return 100;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop3

{

class project

{

public string projectname;

public int duration;

}

}

CONSOLE\_OOP\_INHERITANCE

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_inheritance

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("enter customer details");

Console.WriteLine("ennter id");

int customerid=Convert.ToInt32(Console.ReadLine());

Console.WriteLine("ennter NAME");

String customername = Console.ReadLine();

Console.WriteLine("enter contact number");

string contactnumber=Console.ReadLine();

Console.WriteLine("enter contact mail");

string customeremail = Console.ReadLine();

Console.WriteLine("are you normal or special");

string type = Console.ReadLine();

if (type == "special")

{

Console.WriteLine("enter payment type");

string paymenttype = Console.ReadLine();

Console.WriteLine("enter the deliveryaddress");

string deliveryaddress = Console.ReadLine();

customerspecial obj1 = new customerspecial(customerid, customername, contactnumber, customeremail, paymenttype, deliveryaddress);

string namespecial = obj1.getdetail();

Console.WriteLine(namespecial);

string customerspecial = obj1.getpaymentaandaddress();

Console.WriteLine(customerspecial);

}

else

{

customer obj = new customer(customerid, customername, contactnumber, customeremail);

string name = obj.getdetail();

Console.WriteLine(name);

}

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_inheritance

{

class customer

{

protected int customerid;

protected string customername; // deeriving so protected

protected string contactnumber;

protected string customeremail;

public customer(int customerid,string customername,string contactnumber, string customeremail)

{

this.customerid = customerid;

this.customername = customername;

this.contactnumber = contactnumber;

this.customeremail = customeremail;

Console.WriteLine("customer object creared");

}

public string getdetail()

{

return this.customerid + " " + this.customername + " " + this.customeremail + " " + this.contactnumber;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_inheritance

{

class customerspecial : customer

{

protected string paymenttype;

protected string deliveryadd;

public customerspecial(int customerid, string customername, string contactnumber,

string customeremail, string paymenttype, string deliveryadd) : base(customerid,customername,contactnumber,customeremail)

{

this.paymenttype = paymenttype;

this.deliveryadd = deliveryadd;

Console.WriteLine("derived object creared");

}

public string getpaymentaandaddress()

{

return paymenttype + " " + deliveryadd;

}

}

}

CONSOLE\_OVERRIDE

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Consoleoverrid

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("enteremployee details:");

Console.WriteLine("enteremployee id:");

int employeeid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("enteremployee name:");

string employeename = Console.ReadLine();

Console.WriteLine("enteremployee salaryperday:");

int employeesalaryperday = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("enteremployee no of work days:");

int noofdays= Convert.ToInt32(Console.ReadLine());

Console.WriteLine(" enter the employee type noraml / contract");

string option = Console.ReadLine();

employee obj=null;

if (option == "normal")

{

obj = new employee(employeeid, employeename, employeesalaryperday, noofdays);

}

else if (option == "contract")

{

obj = new employee\_contract(employeeid, employeename, employeesalaryperday, noofdays);

}

else {

Console.WriteLine("not valid");

}

if (obj != null)

{

string details = obj.getdetails();

Console.WriteLine(details);

obj.getwork();

int salary = obj.getsalary();

Console.WriteLine(salary);

}

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Consoleoverrid

{

class employee

{

protected int employeeid;

protected string employeename;

protected int employeeperdaysalary;

protected int noofdays;

public employee(int employeeid, string employeename, int employeeperdaysalary, int noofdays)

{

this.employeeid = employeeid;

this.employeename = employeename;

this.employeeperdaysalary = employeeperdaysalary;

this.noofdays = noofdays;

}

public string getdetails()

{

return this.employeeid + " " + this.employeename;

}

public void getwork()

{

Console.WriteLine("work called");

}

public virtual int getsalary() //giving permission to override so virtual key word

{

int bonus = 5000;

int pf = 2000;

return (noofdays \* employeeperdaysalary) + bonus - pf;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Consoleoverrid

{

class employee\_contract :employee

{

public employee\_contract(int employeeid,string employeename,int employeeperdaysalary,int noofdays):base(employeeid,employeename,employeeperdaysalary,noofdays)

{

Console.WriteLine(" called");

}

public override int getsalary() //being overided from ome othr class so given overruide

{

return (employeeperdaysalary \* noofdays);

}

}

}

ASSIGNMENT\_3.1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assignment\_3.\_1

{

class Program

{

static void Main(string[] args)

{

string option1;

int customerordercount=0;

int ordercount = 0;

int ordervalue=0;

do

{

Console.WriteLine("Cuustomer is there yes? else no");

option1 = Console.ReadLine();

if (option1 == "yes")

{

string option;

int itemid;

int itemqty;

int itemprice = 0;

string customername;

Console.WriteLine("enter the customer name");

customername = Console.ReadLine();

do

{

Console.WriteLine("enter yes to order / no to exit");

option = Console.ReadLine();

if (option == "yes")

{

Console.WriteLine("enter the item to be purchased");

Console.WriteLine("1.phone,2.laptop,3.tab");

itemid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("enter the number of items to be purchased");

itemqty = Convert.ToInt32(Console.ReadLine());

switch (itemid)

{

case 1:

itemprice = 100;

break;

case 2:

itemprice = 200;

break;

case 3:

itemprice = 300;

break;

}

// Console.WriteLine("your order id for the purchased");

order obj = new order(itemid, itemqty, itemprice, customername);

ordercount = order.GetTotalOrderCount();

ordervalue = obj.pordervalue;

}

else

{

Console.WriteLine("thank you !!!!Visit again");

}

} while (option!= "no");

Console.WriteLine("total Number of order you made");

Console.WriteLine("order count" +ordercount);

Console.WriteLine(ordervalue);

Console.WriteLine(order.pcustomercount);

}

else

{

Console.WriteLine(" over");

}

} while (option1 != "no");

Console.WriteLine("total orders made by all customers");

Console.WriteLine(order.GetTotalOrderCount());

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assignment\_3.\_1

{

class order

{

private int orderid;

private static int customercount;

public static int pcustomercount

{

get

{

return customercount;

}

}

private int itemid;

private int itemqty;

private int itemprice;

private static int count;

private string customername;

private static string customernamecheck="";

public int porderid

{

get

{

return orderid;

}

}

public int pordervalue

{

get

{

return itemprice \* itemqty;

}

}

public order(int itemid, int itemqty, int itemprice, string customername)

{

count++;

this.orderid = count;

this.itemid = itemid;

this.itemqty = itemqty;

this.itemprice = itemprice;

this.customername = customername;

if (customernamecheck != customername)

{

customernamecheck = customername;

customercount = 1;

}

else

{

customercount++;

}

// Console.WriteLine("caled constructor");

}

public static int GetTotalOrderCount()

{

//Console.WriteLine("before" + orderid);

// Console.WriteLine("before" + orderid);

return count;

}

}

}

ASSIGNMENT 3.2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assignment3.\_2

{

class Program

{

static void Main(string[] args)

{

manager m1 = manager.getobject();

manager m2 = manager.getobject();

if (m1 == m2)

{

Console.WriteLine("singleton");

}

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assignment3.\_2

{

class manager

{

private int managerid;

private string managername;

public int pmanagerid

{

get

{

return managerid;

}

}

public string pmanagername

{

get

{

return managername;

}

}

private manager()

{

}

static manager m1;

public static manager getobject()

{

if (m1 == null)

{

m1 = new manager();

m1.managerid = 1;

m1.managername = "Aishwarya";

return manager.m1;

}

return manager.m1;

}

}

}

ASSIGNMENT 3.3

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assign3.\_3

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("enter order details");

Console.WriteLine("enter customer id");

int customerid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("enter customer name");

string customername = Console.ReadLine();

Console.WriteLine("enter item price");

int itemprice = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("enter item qty");

int itemqty = Convert.ToInt32(Console.ReadLine());

Console.WriteLine(" enter order or overseas");

string option = Console.ReadLine();

order obj=null;

if(option=="order")

{

obj= new order(customerid, customername, itemqty, itemprice);

}

else if (option=="overseas")

{

obj=new order\_overseas(customerid, customername, itemqty, itemprice);

}

double result = obj.getordervalue();

Console.WriteLine(obj.pcustomerid + " " + obj.pcustomername + " " + result);

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assign3.\_3

{

class order

{

protected int customerid;

protected string customername;

protected int itemqty;

protected int itemprice;

public int pcustomerid

{

get {

return customerid;

}

}

public string pcustomername

{

get

{

return customername;

}

}

public order(int customerid, string customername, int itemqty, int itemprice)

{

this.customerid = customerid;

this.customername = customername;

this.itemprice = itemprice;

this.itemqty = itemqty;

}

public virtual double getordervalue()

{

return itemqty\*itemprice;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assign3.\_3

{

class order\_overseas : order

{

public order\_overseas(int customerid,string customername, int itemqty,int itemprice):base(customerid,customername,itemqty,itemprice)

{

Console.WriteLine("overseas constructor called");

}

public override double getordervalue()

{

return (itemqty\*itemprice)+((itemqty\*itemprice)\*0.1);

}

}

}

CONSOLE\_ABSTRACTCLASS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_abstractclass

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("enter the customer details");

Console.WriteLine("enter the account id");

int accountid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("enter the account balance");

int accountbalance = Convert.ToInt32(Console.ReadLine());

Console.WriteLine(" enter customer name");

String customername = Console.ReadLine();

Console.WriteLine("enter account type");

string Accounttype = Console.ReadLine();

account obj=null;

switch (Accounttype)

{

case "savings":

obj = new savings(accountid, accountbalance, customername);

break;

case "current":

obj = new current(accountid, accountbalance, customername);

break;

case "salary":

obj = new salary(accountid, accountbalance, customername);

break;

default:

Console.WriteLine("enter correctt type");

break;

}

if (obj != null)

{

int balance = obj.getbalance();

Console.WriteLine("account balance" + balance);

Console.WriteLine("enter the amt to withdrawn");

int amount = Convert.ToInt32(Console.ReadLine());

obj.whithdraw(amount);

balance = obj.getbalance();

Console.WriteLine("account balance" + balance);

Console.WriteLine("enter the amt to deposit");

int amoubt = Convert.ToInt32(Console.ReadLine());

obj.deposit(amoubt);

balance = obj.getbalance();

Console.WriteLine("Account balance: " + balance);

Console.WriteLine("enter the chequeno to stop transcation");

string chequeno = Console.ReadLine();

obj.stoppayment(chequeno);

}

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_abstractclass

{

abstract class account

{

protected int accountid;

protected int accountbalance;

protected string customername;

public account ( int accountid,

int accountbalance,

string customername)

{

this.accountid = accountid;

this.accountbalance = accountbalance;

this.customername = customername;

Console.WriteLine("account class constructor called");

}

public int getbalance()

{

return this.accountbalance;

}

public void getstatements()

{

Console.WriteLine("accouunt status");

}

public void blockaccount()

{

Console.WriteLine("accouunt blocked");

}

public abstract bool whithdraw(int amount); // abstract functionp

public abstract bool deposit(int amoubt); //abstarct func

public abstract bool stoppayment(string chequeno); //abstract func

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_abstractclass

{

class savings : account

{

public savings(int accountid,

int accountbalance,

string customername) : base(accountid, accountbalance, customername)

{

Console.WriteLine("saving constructor");

}

public override bool whithdraw(int amount)

{

accountbalance = accountbalance - amount-50;

return true;

}

public override bool deposit(int amoubt)

{

accountbalance = accountbalance + amoubt+ 50;

return true;

}

public override bool stoppayment(string chequeno)

{

Console.WriteLine("cheque stopped"+ chequeno);

return true;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_abstractclass

{

class current :account

{

public current(int accountid, int accountbalance, string customername)

: base(accountid, accountbalance, customername)

{

Console.WriteLine("current account abject created");

}

public override bool whithdraw(int amount)

{

accountbalance = accountbalance - amount;

return true;

}

public override bool deposit(int amoubt)

{

accountbalance = accountbalance + amoubt;

return true;

}

public override bool stoppayment(string chequeno)

{

Console.WriteLine("Cheque is stopped" + chequeno);

return true;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_abstractclass

{

class salary :account

{

public salary(int accountid, int accountbalance, string customername)

: base(accountid, accountbalance, customername)

{

Console.WriteLine("salary constructor called");

}

public override bool whithdraw(int amount)

{

accountbalance = accountbalance - amount - 100;

return true;

}

public override bool deposit(int amoubt)

{

accountbalance = accountbalance + amoubt+8;

return true;

}

public override bool stoppayment(string chequeno)

{

Console.WriteLine("cheque transcation stopped:" + chequeno);

return true;

}

}

}

CONSOLE\_INTERFACE

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_interface

{

class Program

{

static void Main(string[] args)

{

producta obja = new producta(1001, "asd", "chennai");

productb objb = new productb(1001, 111, "dsa", "mad");

transport tran = new transport();

iproducttransport p1= obja;

iproducttransport p2 = objb;

tran.getproduct(p1); // iproducttransport p1= obja;

tran.getproduct(p2); // iproducttransport p1 = obja;

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_interface

{

interface iproducttransport

{

string getfromaddress(); // no access specifier

string gettoaddress(); //no access specifier

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_interface

{

class producta : iproducttransport

{

private int productid;

private string customername;

private string customeraddress;

public producta(int productid, string customername, string customeraddress)

{

this.productid = productid;

this.customername = customername;

this.customeraddress = customeraddress;

}

public void start()

{

Console.WriteLine("start");

}

public void stop()

{

Console.WriteLine("stop");

}

public string getfromaddress()

{

return "company name ";

}

public string gettoaddress()

{

return this.customeraddress;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_interface

{

class productb: iproducttransport

{

private int productid;

private int productprice;

private string customername;

private string deliveryaddress;

public productb(int productid, int productprice, string customername, string deliveryaddress)

{

this.productid = productid;

this.productprice = productprice;

this.customername = customername;

this.deliveryaddress = deliveryaddress;

}

public string getdetails()

{

return this.productid + " " + this.productprice;

}

public string getfromaddress()

{

return "company name";

}

public string gettoaddress()

{

return this.deliveryaddress;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_interface

{

class transport

{

public void getproduct(iproducttransport product)

{

string fromaddress = product.getfromaddress(); ;

string toaddress = product.gettoaddress();

Console.WriteLine("from address :"+fromaddress);

Console.WriteLine(" to address" +toaddress);

}

}

}

CONSOLE\_GC

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_gc

{

class Program

{

static void Main(string[] args)

{

employee obj

using (employee obj = new employee())

{

IDisposable d = obj; //using internally do the folowing two line

d.Dispose();

}

/\* int i = 0;

while (i < 5)

{

test obj = new test();

if (i == 3)

{

GC.SuppressFinalize(obj);

}

obj = null;

i++;

}

GC.Collect(); // destructor called \*/

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_gc

{

class test

{

public test()

{

Console.WriteLine("object creatd");

}

~test() //destructor no parameter // finalize

{

Console.WriteLine("object destroyed");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_oop\_gc

{

class employee :IDisposable

{

public void Dispose()

{

Console.WriteLine("dispose");

//database.close()

//file.close()

}

}

}

CONSOLE\_APPLICATION1 \_INTERFACE

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

HR hr = new HR();

Account ac = new Account();

Manager ma = new Manager();

Employee e = new Employee(123,"aish","pune","kk nagar ",20000,".net ",2,123654789,"hdfc",21);

hr.getemployee(e);

ac.getemployee(e);

ma.getemployee(e);

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Employee :IHRemployee,IAccountEMP,IManagerEmp

{

protected int employeeid;

protected string employeename;

protected string employeecity;

protected string employeeaddress;

protected int employeesalary;

protected string employeeprojectdetails;

protected int employeeexp;

protected int employeeaccountno;

protected string employeebankaccname;

protected int employeeage;

public Employee(int employeeid, string employeename, string employeecity, string employeeaddress, int employeesalary,

string employeeprojectdetails, int employeeexp, int employeeaccountno, string employeebankaccname, int employeeage)

{

this.employeeid = employeeid;

this.employeename = employeename;

this.employeecity = employeecity;

this.employeeaddress = employeeaddress;

this.employeesalary = employeesalary;

this.employeeprojectdetails = employeeprojectdetails;

this.employeeexp = employeeexp;

this.employeeaccountno = employeeaccountno;

this.employeebankaccname = employeebankaccname;

this.employeeage = employeeage;

}

public string GetEmployeeAddress()

{

return this.employeeaddress;

}

public int GetEmployeeSalary()

{

return this.employeesalary;

}

public int GetEmployeeId()

{

return this.employeeid;

}

public int GetEmployeeAccno()

{

return this.employeeaccountno;

}

public int GetEmployeeExperience()

{

return this.employeeexp;

}

public string GetEmployeeProjDetails()

{

return this.employeeprojectdetails;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Manager

{

public void getemployee(IManagerEmp Man)

{

int employeeid = Man.GetEmployeeId();

int employeeexp = Man.GetEmployeeExperience();

string projectdetails = Man.GetEmployeeProjDetails();

Console.WriteLine("employee id " + employeeid+ " employeeexperience " + employeeexp

+ "employee project details " + projectdetails);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Account

{

public void getemployee(IAccountEMP ACC)

{

int employeesalary = ACC.GetEmployeeSalary();

int employeeaccno = ACC.GetEmployeeAccno();

int employeeid = ACC.GetEmployeeId();

Console.WriteLine("employee account num " + employeeaccno + " employeesalary " + employeesalary

+ "employee id " + employeeid);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class HR

{

public void getemployee(IHRemployee HR1)

{

string employeeaddress = HR1.GetEmployeeAddress();

int employeesalary = HR1.GetEmployeeSalary();

int employeeid = HR1.GetEmployeeId();

Console.WriteLine("employee address " + employeeaddress + " employeesalary " + employeesalary

+ "employee id " + employeeid);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

interface IManagerEmp

{

int GetEmployeeId();

int GetEmployeeExperience();

string GetEmployeeProjDetails();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

interface IAccountEMP

{

int GetEmployeeSalary();

int GetEmployeeAccno();

int GetEmployeeId();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

interface IHRemployee

{

string GetEmployeeAddress();

int GetEmployeeSalary();

int GetEmployeeId();

}

}

CONSOLE\_GENERICS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Collections; // for colelctions ARRAYLIST

namespace Console\_generics

{

class Program

{

static void Main(string[] args)

{

Dictionary<string, int> markskeyvalues = new Dictionary<string, int>();

markskeyvalues.Add("a1", 10);

markskeyvalues.Add("a2", 20); // key must be unique

markskeyvalues.Add("a3", 30);

int marks1 = markskeyvalues["a2"]; //addressing through key

Console.WriteLine(marks1);

List<int> list\_marks = new List<int>();

int m1 = 10;

int m2 = 10;

list\_marks.Add(m1);

list\_marks.Add(m2);

int x1 = list\_marks[0];

Console.WriteLine(x1);

for (int i = 0; i < list\_marks.Count; i++)

{

int x2 = list\_marks[i];

Console.WriteLine(x2);

}

foreach (int m in list\_marks)

{

Console.WriteLine(m);

}

List<string> list\_names = new List<string>();

/\* ArrayList list\_marks = new ArrayList();

int m1 = 52; // PLACED INSTACK

int m2 = 22; // PLACED IN STCAK

list\_marks.Add ( m1); // index is 0

list\_marks.Add(m2); // index is 1

list\_marks.Add(44);// index is 2 // BOXING

list\_marks.Remove(m1); // reoves m1 and updates the array now 0 index is 22

bool status1 = true;

bool status2 = false;

Console.WriteLine("count is " + list\_marks.Count);

list\_marks.Add(status1);

list\_marks.Add(status2);

list\_marks.Add("hello");

Console.WriteLine("count is " + list\_marks.Count);

int x1 =Convert.ToInt32( list\_marks[0]); // this is not an array its a collection shoeing 0 as its index

// UNBOXINH

Console.WriteLine("colection" + x1);

test<int> obj = new test<int>();

int tt =obj.getdata(100); // normal func

int t = obj.Genericfun<int>(100);

string s = obj.Genericfun<string>("abc");

Console.WriteLine(t + " " + s + " "+tt); \*/

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_generics

{

class test<x>

{

public x getdata(x i)

{

return i;

}

public T Genericfun<T>(T obj)

{

return obj;

}

}

}

CONSOLE\_GENERIC\_STUDENT

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace console\_generic\_student

{

class Program

{

static void Main(string[] args)

{

college collegeobj = new college("inautix", "chenai");

Console.WriteLine(collegeobj.getcollegedetails());

Console.WriteLine("1. add student 2.find student 3.remove student 4.showstudent 5.showcount 6.exit 7.request a leave");

bool flag = true;

while (flag)

{

Console.WriteLine("enter option");

int option = Convert.ToInt32(Console.ReadLine());

switch (option)

{

case 1:

Console.WriteLine("enter student name");

string studentname = Console.ReadLine();

Console.WriteLine("enter student gender");

string studentgender = Console.ReadLine();

Console.WriteLine("enter student city");

string studentcity = Console.ReadLine();

student stuobj = new student(studentname, studentcity, studentgender);

if (collegeobj.addstudent(stuobj))

{

Console.WriteLine("student sucessfully added");

Console.WriteLine("student id " + stuobj.pstudentid);

}

else

{

Console.WriteLine("student not addded");

}

break;

case 2:

Console.WriteLine("enter student id");

int studentid = Convert.ToInt32(Console.ReadLine());

student foundobj = collegeobj.findstudent(studentid);

if (foundobj != null)

{

Console.WriteLine("found details " + foundobj.getdetails());

}

else

{

Console.WriteLine("student not found");

}

break;

case 3:

Console.WriteLine("enter student id to remove");

int remstuid = Convert.ToInt32(Console.ReadLine());

bool remobj = collegeobj.removestudent(remstuid);

if (remobj == true)

{

Console.WriteLine("student removed ");

}

else

{

Console.WriteLine("student not found");

}

break;

case 4:

collegeobj.showstudents();

break;

case 5:

int count = collegeobj.gettotalstuden();

Console.WriteLine("total count " + count);

break;

case 6:

flag = false;

Console.WriteLine("thank you");

break;

case 7:

Console.WriteLine("enter student id");

int id = Convert.ToInt32(Console.ReadLine());

student obj\_student = collegeobj.findstudent(id);

if (obj\_student != null)

{

Console.WriteLine("enter the reason");

string reason = Console.ReadLine();

obj\_student.take\_leave(reason);

}

else

{

Console.WriteLine("student id incorrect");

}

break;

default:

Console.WriteLine("enter the correct option");

break;

}

}

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace console\_generic\_student

{

class student

{

public delegate void del\_leave(int studentid, string reason);

public event del\_leave evt\_leave;

public void take\_leave(string reason)

{

if (evt\_leave != null)

{

evt\_leave(this.studentid, reason);

Console.WriteLine("event fired");

}

}

private static int count;

private int studentid;

private string studentname;

private string studentcity;

private string studentgender;

public student(string studentname, string studentcity, string studentgender)

{

this.studentcity = studentcity;

this.studentgender = studentgender;

this.studentname = studentname;

count++;

this.studentid = count;

}

public string getdetails()

{

return this.studentid + " " + this.studentname + " " + this.studentcity + " " + this.studentgender;

}

public int pstudentid

{

get {

return this.studentid;

}

}

public string pstudentname

{

get

{

return this.studentname;

}

}

public string pstudentcity

{

get

{

return this.studentcity;

}

}

public string pstudentgender

{

get

{

return this.studentgender;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace console\_generic\_student

{

class college

{

public void leavenotify(int studentid, string reason) // delegate subscriber side

{

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

Console.WriteLine("student request for leavenotify ");

Console.WriteLine("studentid"+ studentid);

Console.WriteLine("reason "+reason);

}

private string collegename;

private string collegeaddress;

private List<student> list\_students = new List<student>();

public college(string collegename, string collegeaddress)

{

this.collegeaddress = collegeaddress;

this.collegename = collegename;

}

public string getcollegedetails()

{

return this.collegename + " " + this.collegeaddress;

}

public bool addstudent(student stdobj)

{

if (stdobj != null)

{

student.del\_leave delobj = new student.del\_leave(this.leavenotify);

stdobj.evt\_leave += delobj;

list\_students.Add(stdobj);

return true;

}

else

{

return false;

}

}

public bool removestudent(int studentid)

{

foreach (student st in list\_students)

{

if (st.pstudentid == studentid)

{

list\_students.Remove(st);

return true;

}

}

return false;

}

public student findstudent(int studentid)

{

foreach (student st in list\_students)

{

if (st.pstudentid == studentid)

{

return st;

}

}

return null;

}

public void showstudents()

{

foreach (student st in list\_students)

{

Console.WriteLine(st.getdetails());

}

}

public int gettotalstuden()

{

return list\_students.Count;

}

}

}

WIN\_DAY5

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace Win\_day5

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Frm\_login ());

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace Win\_day5

{

public partial class frm\_controls : Form

{

public frm\_controls()

{

InitializeComponent();

lst\_cities.Items.Add("chennai");

lst\_cities.Items.Add("pune");

lst\_cities.Items.Add("mumbai");

lst\_cities.Items.Add("delhi");

cmb\_departments.Items.Add("it");

cmb\_departments.Items.Add("hr");

cmb\_departments.Items.Add("accounts");

btn\_validate.Enabled = false;

}

private void btn\_validate\_Click(object sender, EventArgs e)

{

if (lst\_cities.Text == "")

{

MessageBox.Show("select a city");

}

else if(cmb\_departments.Text == "")

{

MessageBox.Show("select a dep");

}

else if (chk\_status.Checked == false)

{

MessageBox.Show(" check the status box");

}

else if (rdb\_female.Checked == false && rdb\_male.Checked == false)

{

MessageBox.Show("enter a gender");

}

else

{

MessageBox.Show("you have selected : " + lst\_cities.Text);

MessageBox.Show("you have selected" + cmb\_departments.Text);

string gender="";

if(rdb\_male.Checked)

{

gender= "male";

}

else{

gender="female";

}

MessageBox.Show("you have selected" +gender);

}

}

private void cmb\_departments\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void chk\_status\_CheckedChanged(object sender, EventArgs e)

{

if (chk\_status.Checked)

{

btn\_validate.Enabled = true;

}

else

{

btn\_validate.Enabled = false;

}

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace Win\_day5

{

public partial class Frm\_login /\* class \*/ : Form

{

public Frm\_login() // constructor

{

InitializeComponent();

}

private void btn\_login\_Click(object sender, EventArgs e)

{

if (txt\_loginid.Text == "")

{

MessageBox.Show("enter login id");

}

else if (txt\_pass.Text == "")

{

MessageBox.Show("enter password");

}

else

{

int loginid = Convert.ToInt32(txt\_loginid.Text);

string password = txt\_pass.Text;

if (loginid == 1001 && password == "password123")

{

MessageBox.Show("valid user");

frm\_controls obj = new frm\_controls();

obj.Show();

}

else

{

MessageBox.Show("invalid user");

}

}

}

}

}

ASSIGNMENT\_DAY 5

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assignment\_day5

{

class Program

{

static void Main(string[] args)

{

company com = new company("ÏNAUTIX ","ASCENDAS");

int option;

do{

Console.WriteLine("enter your option 1.add employee 2.search employee 3.remove employee 4.show employee 5. to apply leave 6.to show my leave details 7.to show all leave details 8.exit ");

option= Convert.ToInt32(Console.ReadLine());

switch(option)

{

case 1:

Console.WriteLine("enter employee name");

string employeename= Console.ReadLine();

Console.WriteLine("enter employee city");

string employeecity =Console.ReadLine();

employee emp= new employee(employeename,employeecity);

if( com.addemployee(emp))

{

Console.WriteLine("employee added sucessfully");

Console.WriteLine("employee id :"+ emp.pemployeeid);

}

else

{

Console.WriteLine("employee not added ");

}

break;

case 2:

Console.WriteLine("enter employee id ");

int employeeid= Convert.ToInt32(Console.ReadLine());

employee empfound =com.searchemployee(employeeid);

if (empfound!=null)

{

Console.WriteLine("employee found");

Console.WriteLine( empfound.getdetails());

}

else

{

Console.WriteLine("employee not found");

}

break;

case 3:

Console.WriteLine("enter employee id to be removed ");

int employeeidremove=Convert.ToInt32(Console.ReadLine());

bool emprem = com.removeemployee(employeeidremove);

if (emprem)

{

Console.WriteLine("employee removed");

}

else{

Console.WriteLine("employee not removed ");

}

break;

case 4:

com.showemployee();

break;

case 5:

Console.WriteLine("enter the employee id");

int employeeidleave = Convert.ToInt32(Console.ReadLine());

employee obj\_employee = com.searchemployee(employeeidleave);

if (obj\_employee != null)

{

Console.WriteLine("enter the reason");

string reason = Console.ReadLine();

obj\_employee.takeleave(reason);

}

else

{

Console.WriteLine("employee not found");

}

break;

case 6:

Console.WriteLine("enter your employee id");

int showempid = Convert.ToInt32(Console.ReadLine());

employee obj\_employeemy = com.searchemployee(showempid);

if (obj\_employeemy != null)

{

com.showmyleave(showempid);

}

else

{

Console.WriteLine("ïd not found");

}

break;

case 7:

Console.WriteLine("employee leave details");

com.approval();

com.showallleave();

break;

default:

Console.WriteLine ("thankyou");

break;

}

}while(option!=8);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assignment\_day5

{

class employee

{

public delegate void del\_leave(leave leave);

public event del\_leave evt\_leave;

public void takeleave(string reason)

{

if (evt\_leave != null)

{

leave l = new leave(this.employeeid, reason);

evt\_leave(l);

Console.WriteLine(l.pleaveid);

}

}

string employeename;

int employeeid;

string employeecity;

private static int count;

public employee(string employeename, string employeecity)

{

this.employeename = employeename;

count++;

this.employeeid = count;

this.employeecity = employeecity;

}

public string pemployeename

{

get {

return employeename;

}

}

public int pemployeeid

{

get {

return employeeid;

}

}

public string pemployeecity

{

get {

return employeecity;

}

}

public string getdetails()

{

return this.employeename + " " + this.employeecity;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assignment\_day5

{

class company

{

List<leave> list\_leave = new List<leave>();

public void leavenotify(leave leave)

{

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

Console.WriteLine("employee id " +leave. pemployeeid);

Console.WriteLine("reason" + leave.pemployeereason);

Console.WriteLine(" leave id" + leave.pleaveid);

list\_leave.Add(leave);

}

public void showallleave()

{

foreach (leave l in list\_leave)

{

l.getdetails();

}

}

public void showmyleave(int employeeid)

{

foreach (leave l in list\_leave)

{

if (l.pemployeeid == employeeid)

{

l.getdetails();

}

}

}

public void approval()

{

foreach (leave l in list\_leave)

{

l.pstatus = true;

}

}

int leaveid;

private static int leavecount;

private string companyname;

private string companyaddress;

List<employee> list\_employee = new List<employee>();

public company(string companyname, string companyaddress)

{

this.companyaddress = companyaddress;

this.companyname = companyname;

}

public string getcompanydetails()

{

return this.companyname + " " + this.companyaddress;

}

public bool addemployee(employee emp)

{

if (emp != null)

{

employee.del\_leave delobj = new employee.del\_leave(this.leavenotify);

emp.evt\_leave += delobj;

list\_employee.Add(emp);

return true;

}

else {

return false;

}

}

public employee searchemployee( int employeeid)

{

foreach (employee emp in list\_employee)

{

if (emp.pemployeeid == employeeid)

{

return emp;

}

}

return null;

}

public bool removeemployee(int employeeid)

{

foreach (employee emp in list\_employee)

{

if (emp.pemployeeid == employeeid)

{

list\_employee.Remove(emp);

Console.WriteLine(" employee removed sucessfully");

return true;

}

}

return false;

}

public void showemployee()

{

foreach (employee emp in list\_employee)

{

Console.WriteLine(emp.getdetails());

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assignment\_day5

{

class leave

{

private int employeeid;

private int leaveid;

private static int count;

private string reason;

private DateTime date;

private bool status = false;

public leave(int employeeid, string reason)

{

count++;

this.employeeid = employeeid;

this.reason = reason;

this.leaveid = count;

}

public int pemployeeid

{

get {

return this.employeeid;

}

}

public string pemployeereason

{

get

{

return this.reason;

}

}

public int pleaveid

{

get

{

return this.leaveid;

}

}

public bool pstatus

{

get

{

return this.status;

}

set {

status = value;

}

}

public void getdetails()

{

Console.WriteLine(this.employeeid + " " + this.leaveid + " " + this.reason+" "+this.status);

}

}

}

ASSIGNMENT\_WINDOWS\_DAY5

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace assignment\_windows\_day5

{

public partial class Form1 : Form

{

private List<order> list\_orders = new List<order>();

string payment;

int ordervalue;

public Form1()

{

InitializeComponent();

cmb\_city.Items.Add("CHENNAI");

cmb\_city.Items.Add("TRICHY");

cmb\_city.Items.Add("MADURAI");

cmb\_city.Items.Add("COIMBATORE");

}

private void radioButton3\_CheckedChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

if (txt\_customername.Text == "")

{

MessageBox.Show("enter your name");

}

else if (txt\_itemname.Text == "")

{

MessageBox.Show("enter item name");

}

else if (txt\_itemprice.Text =="")

{

MessageBox.Show("enter the item price");

}

else if (txt\_itemquantity.Text == "")

{

MessageBox.Show("enter item quantity");

}

else if (txt\_deliveryaddress.Text == "")

{

MessageBox.Show("enter delivery address");

}

else if (rdb\_COD.Checked == false && rdb\_NET.Checked== false && rdb\_PAYTM.Checked ==false)

{

MessageBox.Show("select the payment mode");

}

else

{

int quantity= Convert.ToInt32(txt\_itemquantity.Text);

int price= Convert.ToInt32(txt\_itemprice.Text);

string name = txt\_customername.Text;

string address = txt\_deliveryaddress.Text;

string city = cmb\_city.Text;

string itemname =txt\_itemname.Text;

btn\_placeorder.Enabled = true;

if(rdb\_COD.Checked)

{

payment = "COD";

}

else if(rdb\_NET.Checked)

{

payment =" NET BANKING";

}

else if (rdb\_PAYTM.Checked)

{

payment="PAYTM";

}

string paymentoption = payment;

order ord = new order(quantity, price,name,city,address,paymentoption,itemname);

list\_orders.Add(ord);

MessageBox.Show(" YOUR ORDER ID :" + ord.porderid + " TOTAL VALUE " + ord.getordervalue());

}

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void btn\_showorders\_Click(object sender, EventArgs e)

{

dg\_orders.DataSource = null;

dg\_orders.DataSource = list\_orders;

}

private void dg\_orders\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button1\_Click\_1(object sender, EventArgs e)

{

txt\_customername.Text = "";

txt\_deliveryaddress.Text = "";

txt\_itemname.Text = "";

txt\_itemprice.Text = "";

txt\_itemquantity.Text = "";

rdb\_COD.Checked = false;

rdb\_NET.Checked = false;

rdb\_PAYTM.Checked = false;

cmb\_city.Text = "select a city";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace assignment\_windows\_day5

{

class order

{

int orderid;

static int count;

int ordervalue;

private int itemquantity;

private int itemprice;

private string itemname;

private string customername;

private string paymentoption;

private string ordercity;

private string address;

public order(int itemquantity, int itemprice,string customername,string ordercity,string address,string paymentoption,string itemname)

{

this.itemprice = itemprice;

this.itemquantity = itemquantity;

count++;

this.orderid = count;

this.itemname=itemname;

this.ordercity=ordercity;

this.customername=customername;

this.paymentoption=paymentoption;

this.address = address;

}

public int porderid

{

get

{

return this.orderid;

}

}

public int pitemquantity

{

get {

return itemquantity;

}

}

public int pitemprice

{

get

{

return itemprice;

}

}

public string pcustomername

{

get {

return customername;

}

}

public string pitemname

{

get {

return itemname;

}

}

public string pordercity

{

get {

return ordercity;

}

}

public string paddress

{

get

{

return address;

}

}

public string ppaymentoption

{

get {

return paymentoption;

}

}

public int getordervalue()

{

ordervalue = itemquantity \* itemprice;

return ordervalue;

}

}

}

DLL

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace sum\_library

{

public class clac

{

public int getsum(int num1, int num2) // must be public for dll

{

return num1 + num2+200;

}

}

}

WIN\_ASSEMBLY

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using sum\_library;

namespace Win\_assembly

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void btn\_sum\_Click(object sender, EventArgs e)

{

int num1 = Convert.ToInt32(txt\_num1.Text);

int num2 = Convert.ToInt32(txt\_num2.Text);

// refering to library

//reads the lib

sum\_library.clac sumobj = new sum\_library.clac(); // or use the using namespace func

// clac 0=new clac();

int sum =sumobj.getsum(num1, num2); // reflection

MessageBox.Show("total" + sum);

}

}

}

CONSOLE\_DELEGATES

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace console\_delegate

{

class Program

{

static void Main(string[] args)

{

test t = new test();

t.run();

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace console\_delegate

{

class test

{

public delegate void del(string str);

public delegate void del1(int i);

public delegate void delgeneric<t>(t str); // delegate generics

public void getdata(string msg)

{

Console.WriteLine("getdata" + msg);

}

public void setdata(string str)

{

Console.WriteLine("set data "+str);

}

public void run()

{

int x = 100;

del obj = new del(this.getdata);

obj += new del(this.setdata);// adding address

//obj -= new del(this.setdata); // removing address

obj += delegate(string str)

{

Console.WriteLine("anonymous func " + str + " "+ x);

};

obj += new del((s) => Console.WriteLine("lambda expression " + s));

obj += new del((s) =>

{

Console.WriteLine("lambda expression ");

Console.WriteLine("with multiple line " + s);

});

obj("hello"); // calls getdata // single cast (one address only)

}

}

}

CONSOLE\_EVENT\_DELEGATION MODEL

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_event\_delegationmodel

{

class Program

{

static void Main(string[] args)

{

test t = new test();

t.bind();

t.fire();

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_event\_delegationmodel

{

class test

{

public delegate void del(string str); // publisher // delegate func

public event del evt;// event func

public void fire()//firing event

{

if (evt != null)

{

evt("hello from the event");

}

}

public void event\_handler(string str) //subscriber end

{

Console.WriteLine("event handler " + str);

}

public void bind() //binding code

{

del obj = new del(this.event\_handler);

this.evt += obj;

}

}

}

WIN\_EVENT\_DELEGATION

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace Win\_event\_delegates

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

MessageBox.Show("hello");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace Win\_event\_delegates

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

ASSIGNMENT\_ASSEMBLY

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace assignment\_assembly

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void btn\_salary\_Click(object sender, EventArgs e)

{

int perdaysal = Convert.ToInt32(txt\_perdaysal.Text);

int numberofdays = Convert.ToInt32(txt\_numofdays.Text);

calc\_sal.Class1 obj = new calc\_sal.Class1();

int sal = obj.getsalary(perdaysal,numberofdays);

MessageBox.Show("salary : " + sal);

}

}

}

CALC\_SAL

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace calc\_sal

{

public class Class1

{

public int getsalary(int perdaysalary, int numberofdays)

{

return perdaysalary \* numberofdays;

}

}

}

WIN\_THREADS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace Win\_threads

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new frm\_threadpool());

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace Win\_threads

{

public partial class frm\_async : Form

{

public delegate int del\_getsalary(int perdaysalary, int days);

del\_getsalary obj;

public frm\_async()

{

InitializeComponent();

obj = new del\_getsalary(getsalary);

}

private void btn\_salary\_Click(object sender, EventArgs e)

{

int perdaysalary=Convert.ToInt32(txt\_perdaysalary.Text);

int days= Convert.ToInt32(txt\_numberofdays.Text);

string empid=txt\_employeeid.Text;

// int total= obj(100, 10); //sync

obj.BeginInvoke(perdaysalary, days, new AsyncCallback(callback), empid);

}

public delegate void del();

public void callback(IAsyncResult res)

{

int salary =obj.EndInvoke(res);

// MessageBox.Show("salary of employe id: " + res.AsyncState + " salary :" + salary);

del objdel = delegate

{

lbl\_result.Text = "employee id :" + res.AsyncState + "salary " + salary;

};

this.BeginInvoke(objdel);// main thread will run this func

}

public int getsalary(int perdaysalary, int numberofdays)

{

int total = perdaysalary \* numberofdays;

return total;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Threading;

namespace Win\_threads

{

public partial class frm\_threadpool : Form

{

public frm\_threadpool()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

ThreadPool.SetMinThreads(5, 1000);

ThreadPool.SetMaxThreads(10, 1000);

int count = 0;

while (count < 20)

{

ThreadPool.QueueUserWorkItem(new WaitCallback(call), count);

count++;

}

MessageBox.Show("mainthread ");

}

public void call(object obj)

{

int id = Thread.CurrentThread.ManagedThreadId;

MessageBox.Show("thread id : " + id + " ,loop: " + obj);

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Threading;

namespace Win\_threads

{

public partial class frm\_locking : Form

{

object obj = new object();

public frm\_locking()

{

InitializeComponent();

}

int total;

public void calc()

{

//lock (obj)

//{

bool flag = true;

while (flag)

{

if (Monitor.TryEnter(this))

{

int n1 = Convert.ToInt32(txt\_num1.Text);

int n2 = Convert.ToInt32(txt\_num2.Text);

total = n1 + n2;

Thread.Sleep(10000);

Monitor.Exit(this);

flag = false;

MessageBox.Show("total" + total);

}

else

{

MessageBox.Show("object locked ");

}

}

// }

}

private void btn\_thread1\_Click(object sender, EventArgs e)

{

Thread th1 = new Thread(calc);

th1.Start();

}

private void btn\_thread2\_Click(object sender, EventArgs e)

{

Thread th2 = new Thread(calc);

th2.Start();

}

}

}

WIN\_XML

<?xml version="1.0" encoding="utf-8" ?>

<Customers>

<Customerinfo>

<Customerid>1000</Customerid>

<Customername>aishu</Customername>

<Customerage>25</Customerage>

<Customeraddress>

<streetnumber>212</streetnumber>

<city>chennai</city>

<pin>600044</pin>

</Customeraddress>

<customercontact>0432222001</customercontact>

</Customerinfo>

<Customerinfo>

<Customerid>1001</Customerid>

<Customername>elamukil</Customername>

<Customerage>20</Customerage>

<Customeraddress>

<streetnumber> 500</streetnumber>

<city>pune</city>

<pin>60009</pin>

</Customeraddress>

<customercontact>0232155</customercontact>

</Customerinfo>

<customerinfo/>

</Customers>

WIN\_THREAD\_ASSIGNMENT

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Threading;

namespace Win\_threads\_assignment

{

public partial class Form1 : Form

{

public delegate int del\_result(int num1,int num2,string operation);

del\_result obj;

public Form1()

{

InitializeComponent();

obj = new del\_result(getresult);

}

private void btn\_add\_Click(object sender, EventArgs e)

{

int n1 = Convert.ToInt32(txt\_num1.Text);

int n2 = Convert.ToInt32(txt\_num2.Text);

string operation = "add";

string S = n1 + " + " + n2;

obj.BeginInvoke(n1, n2, operation, new AsyncCallback(callback),S);

}

private void btn\_divide\_Click(object sender, EventArgs e)

{

int n1 = Convert.ToInt32(txt\_num1.Text);

int n2 = Convert.ToInt32(txt\_num2.Text);

string operation = "divide";

string S = n1 + " / " + n2;

obj.BeginInvoke(n1, n2, operation, new AsyncCallback(callback),S);

}

private void btn\_multiply\_Click(object sender, EventArgs e)

{

int n1 = Convert.ToInt32(txt\_num1.Text);

int n2 = Convert.ToInt32(txt\_num2.Text);

string S = n1 + " X " + n2;

string operation = "multiply";

obj.BeginInvoke(n1, n2, operation, new AsyncCallback(callback), S);

}

private void btn\_sub\_Click(object sender, EventArgs e)

{

int n1 = Convert.ToInt32(txt\_num1.Text);

int n2 = Convert.ToInt32(txt\_num2.Text);

string operation = "sub";

string S = n1 + " - " + n2;

obj.BeginInvoke(n1, n2, operation, new AsyncCallback(callback),S);

}

public delegate void del();

public void callback(IAsyncResult res)

{

int result = obj.EndInvoke(res);

del objdel = delegate

{

list\_result.Items.Add("RESULT OF "+ res.AsyncState +" : " + result);

};

this.BeginInvoke(objdel);

}

int result;

public int getresult(int num1, int num2, string operation)

{

switch (operation)

{

case "add":

result = num1 + num2;

break;

case "sub":

result = num1 - num2;

break;

case "multiply":

result = num1 \* num2;

break;

case "divide":

result = num1 / num2;

break;

}

return result;

}

}

}

WIN\_XML\_ASSIGNMENT

<?xml version="1.0" encoding="utf-8" ?>

<employeedetails>

<employeeinfo>

<employeeid>100</employeeid>

<employeename>abi</employeename>

<employeeexperience>2</employeeexperience>

<employeeofficeinfo>

<projectname>.net</projectname>

<skillset>programming </skillset>

<projectdesignation> application developer</projectdesignation>

<duration>2years </duration>

</employeeofficeinfo>

</employeeinfo>

<employeeinfo>

<employeeid>101</employeeid>

<employeename>abcd</employeename>

<employeeexperience>5</employeeexperience>

<employeeofficeinfo>

<projectname>java</projectname>

<skillset> finanancial </skillset>

<projectdesignation> tester</projectdesignation>

<duration> 1year </duration>

</employeeofficeinfo>

</employeeinfo>

</employeedetails>

<?xml version="1.0" encoding="utf-8" ?>

<orders>

<ordersinfo>

<orderid>001</orderid>

<customerid> 1234</customerid>

<customername>ABI</customername>

<productinfo>

<productid>100</productid>

<productname>mobile phone</productname>

<productquantity>10</productquantity>

<productprice>100</productprice>

</productinfo>

<productinfo>

<productid>101</productid>

<productname> laptop</productname>

<productquantity> 20</productquantity>

<productprice> 1000</productprice>

</productinfo>

</ordersinfo>

<ordersinfo>

<orderid>002</orderid>

<customerid>1235</customerid>

<customername> bala</customername>

<productinfo>

<productid>100</productid>

<productname>laptop</productname>

<productquantity>1</productquantity>

<productprice>1000</productprice>

</productinfo>

<productinfo>

<productid>101</productid>

<productname>mobile</productname>

<productquantity>1</productquantity>

<productprice>100</productprice>

</productinfo>

</ordersinfo>

</orders>

WIN\_ADO\_DAY1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace Win\_ado\_day1

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new frm\_showcustomers());

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Configuration;

namespace Win\_ado\_day1

{

public partial class frm\_login : Form

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["constr"].ConnectionString);

public frm\_login()

{

InitializeComponent();

}

private void txt\_password\_TextChanged(object sender, EventArgs e)

{

}

private void btn\_login\_Click(object sender, EventArgs e)

{

con.Open();

SqlCommand com\_login = new SqlCommand(@"select count (\*) from customer

where customerid = @customerid and customerpassword= @customerpassword", con);

com\_login.Parameters.AddWithValue("@customerid" , txt\_loginid.Text);

com\_login.Parameters.AddWithValue("@customerpassword", txt\_password.Text);

int count = Convert.ToInt32(com\_login.ExecuteScalar());

con.Close();

if (count > 0)

{

MessageBox.Show("Valid user");

}

else

{

MessageBox.Show("invalid user");

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient; // important

using System.Configuration;//important

namespace Win\_ado\_day1

{

public partial class Form1 : Form

{

SqlConnection con =

new SqlConnection(ConfigurationManager.ConnectionStrings["constr"].ConnectionString);

public Form1()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void btn\_clear\_Click(object sender, EventArgs e)

{

txt\_customerage.Text = "";

cmb\_cities.Text = "";

txt\_customerid.Text = "";

txt\_customername.Text = "";

txt\_customerpassword.Text = "";

}

private void btn\_addcustomer\_Click(object sender, EventArgs e)

{

con.Open();

SqlCommand com\_customers\_insert =

new SqlCommand("insert customer values (@customername,@customerage,@customercityid,@customerpassword)", con);

com\_customers\_insert.Parameters.AddWithValue("@customername", txt\_customername.Text);

com\_customers\_insert.Parameters.AddWithValue("@customerage", txt\_customerage.Text);

com\_customers\_insert.Parameters.AddWithValue("@customercityid", cmb\_cities.SelectedValue);// cityid;

com\_customers\_insert.Parameters.AddWithValue("@customerpassword", txt\_customerpassword.Text);

com\_customers\_insert.ExecuteNonQuery();

SqlCommand com\_customerid= new SqlCommand ("Select @@identity",con);

int customerid= Convert.ToInt32(com\_customerid.ExecuteScalar());

txt\_customerid.Text= customerid.ToString();

con.Close();

MessageBox.Show("cusotmer added , customer id" + customerid);

}

private void Form1\_Load(object sender, EventArgs e)

{

con.Open();

List<city> list\_cities = new List<city>();

SqlCommand com\_read\_cities = new SqlCommand("select \* from cities", con);

SqlDataReader dr\_cities = com\_read\_cities.ExecuteReader();

while (dr\_cities.Read())

{

city c = new city();

c.cityid = dr\_cities.GetInt32(0);

c.cityname = dr\_cities.GetString(1);

list\_cities.Add(c);

}

con.Close();

cmb\_cities.DataSource = list\_cities;

cmb\_cities.DisplayMember = "cityname";

cmb\_cities.ValueMember = "cityid";

}

private void btn\_findcustomer\_Click(object sender, EventArgs e)

{

con.Open();

SqlCommand com\_read\_customer = new SqlCommand

(@"select customerid,customername,customerage,cityname

from customer

join cities on

customer.customercityid=cities.cityid where customerid=@customerid", con);

int customerid = Convert.ToInt32(txt\_customerid.Text);

com\_read\_customer.Parameters.AddWithValue("@customerid", customerid);

SqlDataReader dr\_customer = com\_read\_customer.ExecuteReader();

if (dr\_customer.Read())

{

txt\_customerid.Text = dr\_customer.GetInt32(0).ToString();

txt\_customername.Text = dr\_customer.GetString(1);

txt\_customerage.Text = dr\_customer.GetInt32(2).ToString();

cmb\_cities.Text = dr\_customer.GetString(3);

}

else

{

MessageBox.Show("customer not found");

}

con.Close();

}

private void btn\_updatecustomer\_Click(object sender, EventArgs e)

{

con.Open();

SqlCommand com\_update\_customer = new SqlCommand(@"update customer set

customername = @customername,customercityid =@customercityid

where customerid=@customerid", con);

com\_update\_customer.Parameters.AddWithValue("@customername", txt\_customername.Text);

com\_update\_customer.Parameters.AddWithValue("@customercityid", cmb\_cities.SelectedValue);

com\_update\_customer.Parameters.AddWithValue("@customerid", txt\_customerid.Text);

int count = com\_update\_customer.ExecuteNonQuery();

con.Close();

if (count > 0)

{

MessageBox.Show("customer updated");

}

else {

MessageBox.Show("customer not found");

}

}

private void btn\_deletecustomer\_Click(object sender, EventArgs e)

{

DialogResult result= MessageBox.Show("do you want to delete this customer ?"," delete ?" ,MessageBoxButtons.YesNo);

if (result == DialogResult.Yes)

{

con.Open();

SqlCommand com\_delete\_customer = new SqlCommand(@"delete customer where customerid=@customerid", con);

com\_delete\_customer.Parameters.AddWithValue("@customerid", txt\_customerid.Text);

int count = com\_delete\_customer.ExecuteNonQuery();

con.Close();

if (count > 0)

{

MessageBox.Show("customer deleted");

}

else

{

MessageBox.Show("customer not found");

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Win\_ado\_day1

{

class city

{

public int cityid { get; set; } // when accessing \_cityid id the variable instead of pcityid

public string cityname { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Configuration;

using System.Data.SqlClient;

namespace Win\_ado\_day1

{

public partial class frm\_showcustomers : Form

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["constr"].ConnectionString);

public frm\_showcustomers()

{

InitializeComponent();

}

private void btn\_showcustomers\_Click(object sender, EventArgs e)

{

if (txt\_cityid.Text =="")

{

SqlDataAdapter data = new SqlDataAdapter(@"select \* from customer", con);

DataSet ds = new DataSet();

data.Fill(ds, "cust");

dg\_showcustomers.DataSource = ds.Tables["cust"];

}

else

{

SqlDataAdapter data\_customer = new SqlDataAdapter(@"select \* from customer where customercityid=@cityid", con);

data\_customer.SelectCommand.Parameters.AddWithValue("@cityid", txt\_cityid.Text);

DataSet ds\_customers = new DataSet();

data\_customer.Fill(ds\_customers, "cust");

dg\_showcustomers.DataSource = ds\_customers.Tables["cust"];

}

}

}

}

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<connectionStrings>

<add name ="constr"

connectionString="server=DA5J8RR7T12;database=inautixADOaug2017;integrated security= true;"/>

</connectionStrings>

</configuration>

WIN\_AD0\_DAY1\_ASSIGNMENT

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace win\_ado\_day1\_assignment

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Configuration;

using System.Data.SqlClient;

namespace win\_ado\_day1\_assignment

{

public partial class frm\_findorder : Form

{

SqlConnection con = new

SqlConnection(ConfigurationManager.ConnectionStrings["constr"].ConnectionString);

public frm\_findorder()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void btn\_findorder\_Click(object sender, EventArgs e)

{

con.Open();

SqlCommand find\_order =

new SqlCommand(@"select orders.customerid,items.itemname,orders.itemqty,orders.orderdate,items.itemprice

from orders

join items

on orders.itemid= items.itemid

where orders.orderid=@orderid", con);

int orderid = Convert.ToInt32(txt\_orderid.Text);

find\_order.Parameters.AddWithValue("@orderid", orderid);

SqlDataReader read\_orders = find\_order.ExecuteReader();

if (read\_orders.Read())

{

lbl\_customeridres.Text = read\_orders.GetInt32(0).ToString();

lbl\_itemnameres.Text = read\_orders.GetString(1);

lbl\_itemqtyres.Text = read\_orders.GetInt32(2).ToString();

lbl\_orderdateres.Text = read\_orders.GetDateTime(3).ToString();

lbl\_itempriceres.Text = read\_orders.GetInt32(4).ToString();

}

else

{

MessageBox.Show("orderid not found");

}

con.Close();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace win\_ado\_day1\_assignment

{

class item

{

public int itemid { get; set; }

public string itemname { get; set; }

}

}

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<connectionStrings>

<add name="constr"

connectionString="server=DA5J8RR7T12;database=inautixADOaug2017;integrated security= true;"/>

</connectionStrings>

</configuration>

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Configuration;

namespace win\_ado\_day1\_assignment

{

public partial class Form1 : Form

{

SqlConnection con = new

SqlConnection(ConfigurationManager.ConnectionStrings["constr"].ConnectionString);

public Form1()

{

InitializeComponent();

cmb\_itemquantity.Items.Add("1");

cmb\_itemquantity.Items.Add("2");

cmb\_itemquantity.Items.Add("3");

cmb\_itemquantity.Items.Add("4");

cmb\_itemquantity.Items.Add("5");

}

private void button1\_Click(object sender, EventArgs e)

{

con.Open();

SqlCommand find\_user = new SqlCommand

(@"select count(\*) from customer where customerid=@customerid", con);

int customerid = Convert.ToInt32(txt\_customerid.Text);

find\_user.Parameters.AddWithValue("@customerid", customerid);

int count = Convert.ToInt32(find\_user.ExecuteScalar());

if (count > 0)

{

SqlCommand update\_order = new SqlCommand(@"insert orders values

(@customerid,@itemid,@itemqty,getdate()) ", con);

update\_order.Parameters.AddWithValue("@customerid", txt\_customerid.Text);

update\_order.Parameters.AddWithValue("@itemid", cmb\_itemname.SelectedValue);

update\_order.Parameters.AddWithValue("@itemqty", cmb\_itemquantity.Text);

update\_order.ExecuteNonQuery();

SqlCommand get\_orderid = new SqlCommand("select @@identity", con);

int orderid= Convert.ToInt32(get\_orderid.ExecuteScalar());

txt\_orderid.Text = orderid.ToString();

MessageBox.Show("order placed + order id " + orderid);

}

else

{

MessageBox.Show("invalid customer id");

}

con.Close();

}

private void Form1\_Load(object sender, EventArgs e)

{

con.Open();

List<item> list\_item = new List<item>();

SqlCommand read\_items = new SqlCommand(@"select \* from items",con);

SqlDataReader dr\_readitems = read\_items.ExecuteReader();

while (dr\_readitems.Read())

{

item i = new item();

i.itemid = dr\_readitems.GetInt32(0);

i.itemname = dr\_readitems.GetString(1);

list\_item.Add(i);

}

con.Close();

cmb\_itemname.DisplayMember = "itemname";

cmb\_itemname.ValueMember = "itemid";

cmb\_itemname.DataSource = list\_item;

}

private void cmb\_itemquantity\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void cmb\_itemname\_SelectedIndexChanged(object sender, EventArgs e)

{

con.Open();

SqlCommand show\_price = new SqlCommand

(@"select itemprice from items where itemid=@itemid", con);

show\_price.Parameters.AddWithValue("@itemid",cmb\_itemname.SelectedValue);

SqlDataReader read\_itemname = show\_price.ExecuteReader();

if (read\_itemname.Read())

{

txt\_itemprice.Text = read\_itemname.GetInt32(0).ToString();

}

con.Close();

}

}

}

WIN\_ADO\_DAY2\_DAL

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace Win\_ado\_day1

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new frm\_showcustomers());

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Win\_ado\_day1

{

class customer

{

public int customerid { get; set; }

public string customername { get; set; }

public int customerage { get; set; }

public int customercityid { get; set; }

public string customerpassword { get; set; }

public string customercityname { get; set; }

public customer(string customername, int customerage,

int customercityid, string customerpassword)

{

this.customerid = customerid;

this.customerpassword = customerpassword;

this.customername = customername;

this.customerage = customerage;

this.customercityid = customercityid;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Win\_ado\_day1

{

class city

{

public int cityid { get; set; } // when accessing \_cityid id the variable instead of pcityid

public string cityname { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Data.SqlClient;

using System.Configuration;

namespace Win\_ado\_day1

{

class citiesdal

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["constr"].ConnectionString);

public List<city> getcities()

{

con.Open();

List<city> list\_cities = new List<city>();

SqlCommand com\_read\_cities = new SqlCommand("select \* from cities", con);

SqlDataReader dr\_cities = com\_read\_cities.ExecuteReader();

while (dr\_cities.Read())

{

city c = new city();

c.cityid = dr\_cities.GetInt32(0);

c.cityname = dr\_cities.GetString(1);

list\_cities.Add(c);

}

con.Close();

return list\_cities;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Data.SqlClient;

using System.Configuration;

namespace Win\_ado\_day1

{

class customersdal

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["constr"].ConnectionString);

public bool addcustomer(customer obj)

{

con.Open();

SqlCommand com\_customers\_insert =

new SqlCommand("insert customer values (@customername,@customerage,@customercityid,@customerpassword)", con);

com\_customers\_insert.Parameters.AddWithValue("@customername", obj.customername);

com\_customers\_insert.Parameters.AddWithValue("@customerage", obj.customerage);

com\_customers\_insert.Parameters.AddWithValue("@customercityid", obj.customercityid);// cityid;

com\_customers\_insert.Parameters.AddWithValue("@customerpassword", obj.customerpassword);

com\_customers\_insert.ExecuteNonQuery();

SqlCommand com\_customerid= new SqlCommand ("Select @@identity",con);

int customerid= Convert.ToInt32(com\_customerid.ExecuteScalar());

obj.customerid= customerid;

con.Close();

return true;

}

public customer findcustomer(int customerid)

{

con.Open();

SqlCommand com\_read\_customer = new SqlCommand

(@"select customerid,customername,customerage,cityname

from customer

join cities on

customer.customercityid=cities.cityid where customerid=@customerid", con);

customer obj= null;

com\_read\_customer.Parameters.AddWithValue("@customerid", customerid);

SqlDataReader dr\_customer = com\_read\_customer.ExecuteReader();

if (dr\_customer.Read())

{

int custid = dr\_customer.GetInt32(0);

string custname= dr\_customer.GetString(1);

int custage = dr\_customer.GetInt32(2);

string custcity = dr\_customer.GetString(3);

obj = new customer (custname,custage,0,"NA");

obj.customercityname=custcity;

obj.customerid=custid;

}

con.Close();

return obj;

}

public bool logincustomer(int customerid,string customerpassword)

{

con.Open();

SqlCommand com\_login = new SqlCommand(@"select count (\*) from customer

where customerid = @customerid and customerpassword= @customerpassword", con);

com\_login.Parameters.AddWithValue("@customerid" , customerid);

com\_login.Parameters.AddWithValue("@customerpassword",customerpassword);

int count = Convert.ToInt32(com\_login.ExecuteScalar());

con.Close();

if (count > 0)

{

return true;

}

else

{

return false;

}

}

public bool updatecustomer(customer obj)

{

con.Open();

SqlCommand com\_update\_customer = new SqlCommand(@"update customer set

customername = @customername,customercityid =@customercityid

where customerid=@customerid", con);

com\_update\_customer.Parameters.AddWithValue("@customername",obj.customername);

com\_update\_customer.Parameters.AddWithValue("@customercityid", obj.customercityid);

com\_update\_customer.Parameters.AddWithValue("@customerid", obj.customerid);

int count = com\_update\_customer.ExecuteNonQuery();

con.Close();

if (count > 0)

{

return true;

}

else {

return false;

}

}

public bool deletecustomer(int customerid)

{

con.Open();

SqlCommand com\_delete\_customer = new SqlCommand

(@"delete customer where customerid=@customerid", con);

com\_delete\_customer.Parameters.AddWithValue("@customerid", customerid);

int count = com\_delete\_customer.ExecuteNonQuery();

con.Close();

if (count > 0)

{

return true;

}

else

{

return false;

}

}

public List<customer> getcustomers(int cityid =0)

{

con.Open();

List<customer> list\_customers = new List<customer>();

SqlCommand com\_customers = null;

if (cityid == 0)

com\_customers = new SqlCommand(@"select \* from customer", con);

else

{

com\_customers = new SqlCommand(@" select \* from customer where customercityid=@cityid", con);

com\_customers.Parameters.AddWithValue("@cityid",cityid);

}

SqlDataReader dr\_customers = com\_customers.ExecuteReader();

while (dr\_customers.Read())

{

int customerid = dr\_customers.GetInt32(0);

string customername = dr\_customers.GetString(1);

int customerage = dr\_customers.GetInt32(2);

int customercityid = dr\_customers.GetInt32(3);

string customerpassword = dr\_customers.GetString(4);

customer obj = new customer(customername, customerage, customercityid, customerpassword);

obj.customerid = customerid;

list\_customers.Add(obj);

}

con.Close();

return list\_customers;

}

}

}

WIN\_TRANS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace WIN\_TRANS

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Configuration;

using System.Data.SqlClient;

namespace WIN\_TRANS

{

class ordersinfo\_dal

{

SqlConnection con = new SqlConnection

(ConfigurationManager.ConnectionStrings["constr"].ConnectionString);

public bool placeorder(order ord, List<item> items)

{

con.Open();

SqlTransaction tran = con.BeginTransaction();

SqlCommand com\_order\_insert = new SqlCommand(@"insert ordersinfo values

(@customername,@customeraddress,getdate())", con);

com\_order\_insert.Parameters.AddWithValue("@customername", ord.customername);

com\_order\_insert.Parameters.AddWithValue("@customeraddress", ord.customeraddress);

com\_order\_insert.Transaction = tran;

com\_order\_insert.ExecuteNonQuery();

SqlCommand com\_orderid = new SqlCommand(@"select @@identity", con);

com\_orderid.Transaction = tran;

ord.orderid = Convert.ToInt32(com\_orderid.ExecuteScalar());

SqlCommand com\_getdate = new SqlCommand(@"select orderdate from ordersinfo where orderid =@orderid", con);

com\_getdate.Parameters.AddWithValue("@orderid", ord.orderid);

com\_getdate.Transaction = tran;

ord.orderdate = Convert.ToDateTime(com\_getdate.ExecuteScalar());

foreach (item itemobj in items)

{

SqlCommand com\_items\_insert =

new SqlCommand(@"insert orderdetails values( @orderid,@itemid,@itemqty,@itemprice)", con);

com\_items\_insert.Parameters.AddWithValue("@orderid", ord.orderid);

com\_items\_insert.Parameters.AddWithValue("@itemid", itemobj.itemid);

com\_items\_insert.Parameters.AddWithValue("@itemqty", itemobj.itemqty);

com\_items\_insert.Parameters.AddWithValue("@itemprice", itemobj.itemprice);

com\_items\_insert.Transaction = tran;

com\_items\_insert.ExecuteNonQuery();

}

System.Windows.Forms.MessageBox.Show("wait ");

// tran.Rollback();

tran.Commit();// done no error must error

con.Close();

return true;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace WIN\_TRANS

{

class order

{

public int orderid { get; set; }

public string customername { get; set; }

public string customeraddress { get; set; }

public DateTime orderdate { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace WIN\_TRANS

{

class item

{

public int itemid { get; set; }

public int itemqty { get; set; }

public int itemprice { get; set; }

}

}

WIN\_ADO\_DAY 2\_ASSIGNMENT\_DAL

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace win\_ado\_day2\_assignment

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace win\_ado\_day2\_assignment

{

class tickets

{

public int ticketnumber { get; set; }

public string moviename { get; set; }

public DateTime moviedate { get; set; }

public string timing { get; set; }

public int numberoftickets { get; set; }

public int ticketprice { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Configuration;

using System.Data.SqlClient;

namespace win\_ado\_day2\_assignment

{

class tickets\_dal

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["constr"].ConnectionString);

public transactioninfo buytickets(tickets tick, transactions transac)

{

con.Open();

SqlTransaction tran = con.BeginTransaction();

SqlCommand com\_insert\_transaction = new SqlCommand

(@"insert transactions values (@bankname,@accountnumber,@amount,getdate())", con);

com\_insert\_transaction.Parameters.AddWithValue("@bankname", transac.bankname);

com\_insert\_transaction.Parameters.AddWithValue("@accountnumber", transac.accountnumber);

com\_insert\_transaction.Parameters.AddWithValue("@amount", transac.amount);

com\_insert\_transaction.Transaction = tran;

com\_insert\_transaction.ExecuteNonQuery();

SqlCommand com\_get\_transactionnumber = new SqlCommand(@"select @@identity", con);

com\_insert\_transaction.Transaction=tran;

com\_get\_transactionnumber.Transaction = tran;

transac.transactionnumber = Convert.ToInt32(com\_get\_transactionnumber.ExecuteScalar());

SqlCommand com\_insert\_tickets = new SqlCommand(@"insert tickets values

(@moviename,@moviedate,@timing,@numberoftickets,@ticketprice,@transactionnumber)", con);

com\_insert\_tickets.Parameters.AddWithValue("@moviename", tick.moviename);

com\_insert\_tickets.Parameters.AddWithValue("@moviedate", tick.moviedate);

com\_insert\_tickets.Parameters.AddWithValue("@timing", tick.timing);

com\_insert\_tickets.Parameters.AddWithValue("@numberoftickets", tick.numberoftickets);

com\_insert\_tickets.Parameters.AddWithValue("@ticketprice", tick.ticketprice);

com\_insert\_tickets.Parameters.AddWithValue("@transactionnumber", transac.transactionnumber);

com\_insert\_tickets.Transaction = tran;

com\_insert\_tickets.ExecuteNonQuery();

SqlCommand com\_get\_ticketnumber = new SqlCommand(@"select @@identity", con);

com\_get\_ticketnumber.Transaction = tran;

tick.ticketnumber = Convert.ToInt32(com\_get\_ticketnumber.ExecuteScalar());

System.Windows.Forms.MessageBox.Show("wait transaction under process");

tran.Commit();

con.Close();

return transactioninfo.done;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace win\_ado\_day2\_assignment

{

enum transactioninfo

{

done,failednobalance,failedbankdenied

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace win\_ado\_day2\_assignment

{

class transactions

{

public int transactionnumber { get; set; }

public string bankname { get; set; }

public int accountnumber { get; set; }

public int amount { get; set; }

public DateTime transactiondate { get; set; }

}

}