ASSIGNMENT 6

1.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace delegateassignment1

{

public delegate double ManagerSal(double salary);

public delegate double Food(double FA);

public delegate double Petrol(double PA);

public delegate double Others(double OA);

public delegate double Gross(double GS);

public delegate double Pf(double PF);

public delegate double Netsal(double NS);

class @delegate

{

public double PA;

public double FA;

public double OA;

public double salary;

public double GS;

public double NS;

public double PF;

public double TDS;

public double ManagerSalary(double salary)

{

Console.WriteLine("manager salary" + salary);

return salary;

}

public double FoodAllowances(double Salary)

{

FA = (Salary \* 13) / 100;

Console.WriteLine("Food allowances " + FA);

return FA;

}

public double PetrolAllowances(double Salary)

{

PA = (Salary \* 8) / 100;

Console.WriteLine("petrol allowances " + PA);

return PA;

}

public double OtherAllowances(double Salary)

{

OA = (Salary \* 3) / 100;

Console.WriteLine("Other allowances " + OA);

return OA;

}

public double GrossSalary(double Salary)

{

GS = Salary + FA + PA + OA;

Console.WriteLine("Gross salary" + GS);

return GS;

}

public double CalculateSalary()

{

PF = (GS \* 10) / 100;

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

TDS = (GS \* 18) / 100;

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

NS = GS - (PF + TDS);

Console.WriteLine("NetSalary: " + NS);

return NS;

}

public static void Main(string[] args)

{

@delegate d = new @delegate();

d.FoodAllowances(25000);

d.PetrolAllowances(25000);

d.OtherAllowances(25000);

d.GrossSalary(25000);

d.CalculateSalary();

}

}

}

2.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DelegatesAssignment

{

public delegate void EmployeeDelegate();

public class MCDelegate

{

public static void PetrolAllowance()

{

double salary;

Console.WriteLine("enter salary");

salary = double.Parse(Console.ReadLine());

double PA = (8 \* salary) / 100;

Console.WriteLine("petrol allowance: " + PA);

}

public static void FoodAllowance()

{

double salary;

Console.WriteLine("enter salary");

salary = double.Parse(Console.ReadLine());

double FA = (13 \* salary) / 100;

Console.WriteLine("food allowance : " + FA);

}

public static void OtherAllowances()

{

double salary;

Console.WriteLine("enter salary");

salary = double.Parse(Console.ReadLine());

double OA = (3 \* salary) / 100;

Console.WriteLine("Other allowance: " + OA);

}

public static void Details()

{

int EmpId = 1;

string EmpName = "Navya";

Console.WriteLine("Employee id :" + EmpId);

Console.WriteLine("Employee name: " + EmpName);

}

static void Main(string[] args)

{

MCDelegate M = new MCDelegate();

EmployeeDelegate emp1 = new EmployeeDelegate(PetrolAllowance);

EmployeeDelegate emp2 = new EmployeeDelegate(FoodAllowance);

EmployeeDelegate emp3 = new EmployeeDelegate(OtherAllowances);

EmployeeDelegate emp4 = new EmployeeDelegate(Details);

EmployeeDelegate emp5 = emp1 + emp2 + emp3 + emp4;

emp5();

}

}

}

3.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace delegateassignment2

{

class bank

{

public static double Atm\_Pin { get; private set; }

public delegate void Bank(int x);

public class Account

{

public double AccountNumber { get; set; }

public string CustomerName { get; set; }

public decimal bank\_balance = 13434;

public event Bank UnderBalance;

public event Bank ZeroBalance;

public void Insufficient(int i)

{

UnderBalance(i);

}

public void DepositAmount(int d)

{

ZeroBalance(d);

}

public void withdraw(int i)

{

if (i < bank\_balance && bank\_balance != 0)

{

Console.WriteLine("please take your money");

Console.WriteLine("Transaction Successfull");

Console.WriteLine("Available Balance is:" + (bank\_balance - i));

}

else if (i > bank\_balance && bank\_balance != 0)

{

Console.WriteLine("Insufficient Amount");

Console.WriteLine("Your Current Balance is:" + bank\_balance + "only");

}

else

{

Console.WriteLine("Zero Balance:" + bank\_balance);

}

}

public void deposit(int i)

{

Console.WriteLine("Balance after depositing:" + (bank\_balance + i));

}

}

static void Main(string[] args)

{

Account A = new Account();

Console.WriteLine("Hello Customer!");

Console.WriteLine("Enter Your Pin Number ");

Atm\_Pin = double.Parse(Console.ReadLine());

Console.WriteLine();

Console.WriteLine("\*Welcome to ATM Service\*\n");

Console.WriteLine("1. Withdraw");

Console.WriteLine("2. Deposit");

Console.WriteLine("select your option: 1 or 2 ");

string withdraw = Console.ReadLine();

if (withdraw == "1")

{

Console.WriteLine("Enter your amount to be Withdraw");

int withdrawbalance = int.Parse(Console.ReadLine());

A.UnderBalance += new Bank(A.withdraw);

A.Insufficient(withdrawbalance);

}

else

{

Console.WriteLine("Enter your amount to be Deposit:");

int depositbalance = int.Parse(Console.ReadLine());

A.ZeroBalance += new Bank(A.deposit);

A.deposit(depositbalance);

}

}

}

}

4.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace delegateassign

{

public delegate void Notify1();

class ICICIBank

{

protected double AccountNumber;

protected string CustomerName;

protected double Balance;

protected double Amount;

public event Notify1 ZeroBalance;

public event Notify1 UnderBalance;

public void Diposit()

{

Console.WriteLine("Welcome to ICICI Services");

Console.WriteLine("enter account number");

AccountNumber = double.Parse(Console.ReadLine());

Console.WriteLine("enter customer name");

CustomerName = Console.ReadLine();

Console.WriteLine("enter amount to disposit");

Balance = double.Parse(Console.ReadLine());

Console.WriteLine("balance amount is" + ":" + Balance);

}

public virtual void WithDraw1()

{

Console.WriteLine("enter amount to withdrawl");

Amount = int.Parse(Console.ReadLine());

if (Balance == 0)

{

OnZeroBalance();

}

else if (Amount > Balance)

{

Onunderbalance();

}

else

{

Console.WriteLine("your balance is" + ":" + (Balance - Amount) + ";" + "ICICITRANSACTION SUCCESSFULL");

}

}

protected virtual void OnZeroBalance()

{

ZeroBalance?.Invoke();

}

protected virtual void Onunderbalance()

{

UnderBalance?.Invoke();

}

}

class ICICI

{

public static void Main()

{

ICICIBank account = new ICICIBank();

account.Diposit();

account.WithDraw1();

account.ZeroBalance += z1\_ZeroBalance;

account.UnderBalance += z2\_underbalace;

}

public static void z1\_ZeroBalance()

{

Console.WriteLine("Transaction cannot be continued as balance is zero in Account ");

}

public static void z2\_underbalace()

{

Console.WriteLine("Transaction cannot be continued as balance is insufficient");

}

}

}

5.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace delegateassignmemt4

{

class bank

{

public static double Atm\_Pin { get; private set; }

public delegate void Bank(int x);

public class Account

{

public double AccountNumber { get; set; }

public string CustomerName { get; set; }

public decimal bank\_balance = 13434;

public event Bank UnderBalance;

public event Bank ZeroBalance;

public void Insufficient(int i)

{

UnderBalance(i);

}

public void DepositAmount(int d)

{

ZeroBalance(d);

}

public void withdraw(int i)

{

if (i < bank\_balance && bank\_balance != 0)

{

Console.WriteLine("please take your money");

Console.WriteLine("Transaction Successfull");

Console.WriteLine("Available Balance is:" + (bank\_balance - i));

}

else if (i > bank\_balance && bank\_balance != 0)

{

bank\_balance = bank\_balance - i;

if (bank\_balance > (-1000))

{

Console.WriteLine("Your Current Balance is:" + bank\_balance + "only");

}

else

{

Console.WriteLine("withdraw is not possible");

}

}

else

{

Console.WriteLine("Zero Balance:" + bank\_balance);

}

}

public void deposit(int i)

{

Console.WriteLine("Balance after depositing:" + (bank\_balance + i));

}

}

static void Main(string[] args)

{

Account A = new Account();

Console.WriteLine("Hello Customer!");

Console.WriteLine("Enter Your Pin Number ");

Atm\_Pin = double.Parse(Console.ReadLine());

Console.WriteLine();

Console.WriteLine("\*Welcome to ATM Service\*\n");

Console.WriteLine("1. Withdraw");

Console.WriteLine("2. Deposit");

Console.WriteLine("select your option: 1 or 2 ");

string withdraw = Console.ReadLine();

if (withdraw == "1")

{

Console.WriteLine("Enter your amount to be Withdraw");

int withdrawbalance = int.Parse(Console.ReadLine());

A.UnderBalance += new Bank(A.withdraw);

A.Insufficient(withdrawbalance);

}

else

{

Console.WriteLine("Enter your amount to be Deposit:");

int depositbalance = int.Parse(Console.ReadLine());

A.ZeroBalance += new Bank(A.deposit);

A.deposit(depositbalance);

}

}

}

}