**Assignment No. 6**

1. In Assignment 3 of previous session, print the details of Managers with the help of delegate

EmployeeDelegate. (UniCast Delegate)

**INPUT:**

using System;

namespace DelegateAssignment

{

public delegate void EmployeeDelegate(int ManagerId, string ManagerName,double ManagerSalary);

public class DelegateProgram

{

public static void ManagerDetails(int ManagerId, string ManagerName, double ManagerSalary)

{

Console.WriteLine("Manager Details");

Console.WriteLine("Manager Id: {0}\nManager Name: {1}\nManager Salary:{2}", ManagerId, ManagerName, ManagerSalary);

}

public static void Main(string[] args)

{

EmployeeDelegate delegateemp = new EmployeeDelegate(DelegateProgram.ManagerDetails);

delegateemp(111, "Muskan Khan", 90000);

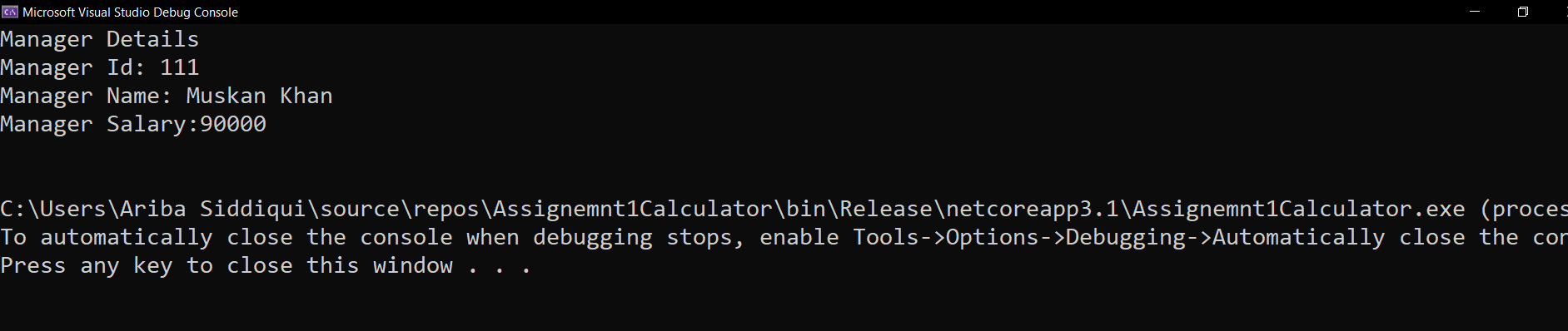
Console.Read();

}

}

}

**OUPUT:**



1. Create a MultiCast Delegate by adding address of another method to the above delegate to list the details of Marketing Executive.

**INPUT:**

using System;

namespace DelegateAssignment

{

public delegate void EmployeeDelegate(int Id, string Name, double Salary);

public class DelegateProgram

{

public static void ManagerDetails(int ManagerId, string ManagerName, double ManagerSalary)

{

Console.WriteLine("Manager Details\n");

Console.WriteLine("Manager Id: {0}\nManager Name: {1}\nManager Salary:{2}\n", ManagerId, ManagerName, ManagerSalary);

}

public static void MarketingExecutive(int MarketingExecuId , string MarketingExecuName, double MarketingExecuSalary)

{

Console.WriteLine("Marketing Executive Details..\n");

Console.WriteLine("Manager Id: {0}\nManager Name: {1}\nManager Salary:{2} \n", MarketingExecuId, MarketingExecuName, MarketingExecuSalary);

}

public static void Main(string[] args)

{

EmployeeDelegate delegateemp = DelegateProgram.ManagerDetails;

delegateemp(111, "Muskan Khan", 90000);

delegateemp(201, "Aidtya Salunke", 100000);

delegateemp(101, "Pratik Hadawale", 97000);

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

delegateemp += DelegateProgram.MarketingExecutive;

Console.WriteLine("");

delegateemp(105, "Sushmita Gupta", 80000);

Console.Read();

}

}

}

**OUTPUT:**



1. Write a console Application for banking domain, having Account class with data members as account number, customer name, balance . It should have Withdraw and Deposit methods for performing banking transaction. It should also define UnderBalance, BalanceZero events. These events would be raised when balance of account is less than certain amount and equal to zero respectively.

**INPUT:**

using System;

namespace BankDomain

{

public delegate void BankDetail();

#region Account Class

class Account

{

public static int AccountNum;

public static string CustomerName;

public static int Balance;

public static event BankDetail UnderBalance;

public static event BankDetail BalanceZero;

static void Main()

{

Console.WriteLine("Enter the action to be performed");

Console.WriteLine("Press 1 for Withdraw");

Console.WriteLine("Press 2 for Deposit");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

try

{

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

switch (action)

{

case 1:

{

Withdraw();

break;

}

case 2:

{

Deposit();

break;

}

default:

Console.WriteLine("Please enter proper number");

break;

}

Console.WriteLine("");

//Events

UnderBalance += new BankDetail(Withdraw);

BalanceZero += new BankDetail(Deposit);

}

catch(Exception ex)

{

Console.WriteLine(ex.Message);

}

}

#region Withdraw Method

static void Withdraw() {

Console.Write("Enter Account Number:");

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

Console.Write("Enter Customer Name:");

CustomerName = Console.ReadLine();

Console.Write("Enter Balance:");

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

try

{

if (Balance < 500)

{

Console.WriteLine("Balance Zero: Insufficient Balance");

}

else

{

Console.WriteLine("Amount {0} Withdraw Successfully..", Balance);

}

}

catch(Exception ex)

{

Console.WriteLine(ex.Message);

}

}

#endregion

#region Deposit Method

static void Deposit() {

Console.Write("Enter Account Number:");

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

Console.Write("Enter Customer Name:");

CustomerName = Console.ReadLine();

Console.Write("Enter Balance:");

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

try

{

if (Balance > 1000)

{

Console.WriteLine("Amount Deposited Successfully...");

}

else

{

Console.WriteLine("This amount is not sufficeint ");

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

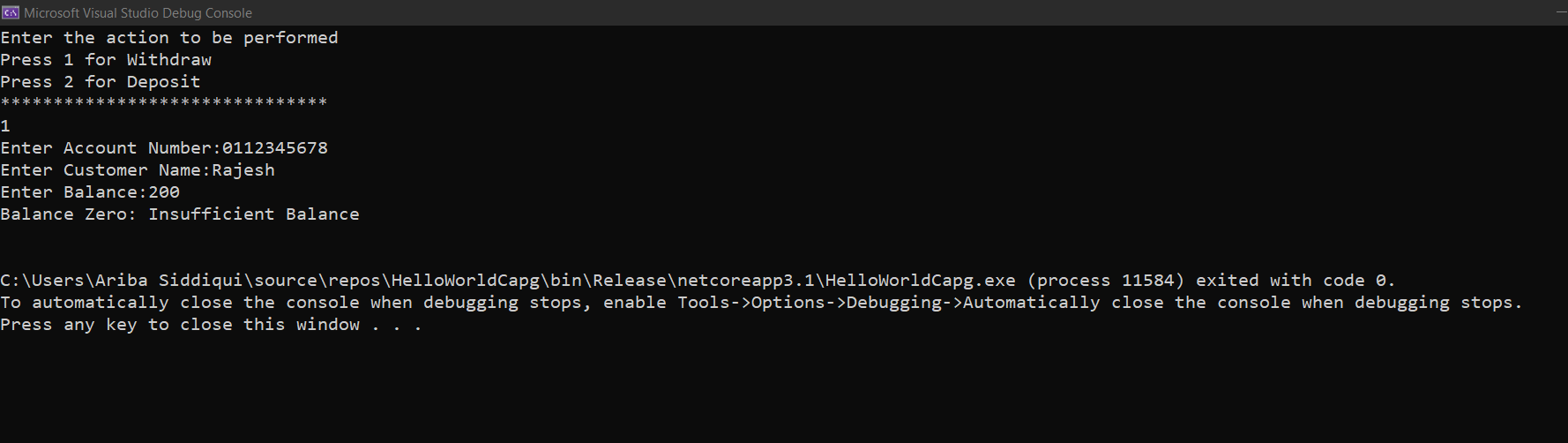
#endregion

}

#endregion

}

**Balance Zero Event Output:**



**Under Balance Event Output:**

