ASSIGNMENT 7

1)

using System;

using System.IO;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FileIO\_Assignment

{

class Program

{

static void Main(string[] args)

{

try

{

string text = System.IO.File.ReadAllText(@"C:\test\text files\ritika.txt");

System.Console.WriteLine("Contents of text file = {0} ", text);

string[] lines = System.IO.File.ReadAllLines("C:\test\text files\ritika.txt");

System.Console.WriteLine("Contents of text file2 = ");

foreach (string line in lines)

{

Console.WriteLine("\t " + line);

}

Console.WriteLine("press any key to exit.");

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

string rootdir = @"C:\test\text files";

string[] files = Directory.GetFiles(rootdir);

Console.WriteLine(string.Join(Environment.NewLine, files));

string[] dirs = Directory.GetDirectories(rootdir);

string[] subdir = Directory.GetDirectories(rootdir);

Console.WriteLine(string.Join(Environment.NewLine, dirs));

DirectoryInfo directoryInfo = new DirectoryInfo(rootdir);

if (directoryInfo.Exists)

{

Console.WriteLine("Directory already exsits");

}

directoryInfo.Create();

Console.WriteLine("Directory created");

FileInfo fileinfo = new FileInfo(@"C:\test\text files\ritika.txt");

StreamWriter streamWriter = fileinfo.CreateText();

streamWriter.WriteLine("file created");

streamWriter.WriteLine("welcome to c#");

streamWriter.Close();

StreamReader streamReader = fileinfo.OpenText();

Console.WriteLine(streamReader.ReadToEnd());

streamReader.Close();

System.Console.ReadKey();

}

}

}

2)

using System;

using System.IO;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace File.Io\_assignment

{

public class StreamReaderWriter

{

static void Main(string[] args)

{

Account account = new Account();

Console.WriteLine("Deposit");

account.Deposit();

Console.WriteLine("Reading the data from file");

Console.WriteLine();

account.ReaderFromFile();

Console.WriteLine("Withdraw");

account.Withdraw();

Console.WriteLine("Reading the data from file");

Console.WriteLine();

account.ReaderFromFile();

}

}

class Account

{

private double balance = 20000;

String bank = String.Empty;

String name = String.Empty;

int account;

double withdraw, deposit, totalbal;

String myPath = @"E:\Directory\Text.txt";

public void Deposit()

{

string temp = "";

StreamWriter writer = new StreamWriter(myPath, true);

do

{

Console.WriteLine("If you want Enter Or Exit Type: y/n ");

temp = Console.ReadLine().ToLower();

if (temp != "n")

{

Console.WriteLine("Enter Customer name:");

name = Console.ReadLine();

writer.WriteLine(name);

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

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

writer.WriteLine(account);

Console.WriteLine("Enter Deposit Amount :");

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

writer.WriteLine(deposit);

totalbal = balance + deposit;

}

} while (temp != "n");

writer.Close();

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

Console.WriteLine("Account Number: " + account);

Console.WriteLine("Total Amount After Deposit : " + totalbal);

}

public void Withdraw()

{

string temp = "";

StreamWriter sr = new StreamWriter(myPath, true);

do

{

Console.WriteLine("If you want Enter Or Exit Type: y/n ");

temp = Console.ReadLine().ToLower();

if (temp != "n")

{

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

name = Console.ReadLine();

sr.WriteLine(name);

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

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

sr.WriteLine(account);

Console.WriteLine("Enter Withdraw Amount :");

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

sr.WriteLine(withdraw);

if (withdraw <= balance)

{

totalbal = balance - withdraw;

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

Console.WriteLine("Account Number: " + account);

Console.WriteLine("Amount after Withdraw : " + totalbal);

}

else

Console.WriteLine("\n\nUnder balance");

}

} while (temp != "n");

sr.Close();

}

public void ReaderFromFile()

{

StreamReader streamReader = new StreamReader(myPath);

while (!streamReader.EndOfStream)

{

Console.WriteLine(streamReader.ReadLine());

}

streamReader.Close();

}

}

}

3)

using System;

using System.IO;

using System.Runtime.Serialization;

using System.Runtime.Serialization.Formatters.Binary;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace File\_IO

{

[Serializable]

class Employee

{

public int EMPID;

public string Name;

public double Salary, GrossSalary, PF, TDS, NetSalary;

public void employee()

{

Console.WriteLine("emter empid");

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

Console.WriteLine("enter name");

Name = Console.ReadLine();

}

}

[Serializable]

class Manager : Employee

{

public double PetrolAllowance;

public double FoodAllowance;

public double OtherAllowance;

public void manager()

{

Console.WriteLine("enter salary ");

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

PetrolAllowance = (8 \* Salary) / 100;

Console.WriteLine("PetrolAllowance allowances:" + PetrolAllowance);

FoodAllowance = (13 \* Salary) / 100;

Console.WriteLine("food allowances: " + FoodAllowance);

OtherAllowance = (3 \* Salary) / 100;

Console.WriteLine("other allowances: " + OtherAllowance);

GrossSalary = Salary + PetrolAllowance + FoodAllowance + OtherAllowance;

Console.WriteLine("GrossSalry of Manager: " + GrossSalary);

PF = (10 \* Salary) / 100;

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

TDS = (18 \* Salary) / 100;

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

}

public void CalculateSalary()

{

NetSalary = GrossSalary - (PF + TDS);

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

}

}

[Serializable]

class MarketingExecutive : Manager

{

double KilometerTravel;

double TourAllowances;

double TelephoneAllowances;

public void marketingExecutive()

{

Console.WriteLine("enter salary of MarketingExecutive");

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

Console.WriteLine("enter kilo meters");

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

TourAllowances = 5 \* KilometerTravel;

Console.WriteLine("TourAllowances allowances: " + TourAllowances);

TelephoneAllowances = 1000;

Console.WriteLine("Telephone allowances: " + TelephoneAllowances);

GrossSalary = Salary + TourAllowances + TelephoneAllowances;

Console.WriteLine("GrossSalary of marketing executive: " + GrossSalary);

PF = (10 \* Salary) / 100;

Console.WriteLine("PF of Marketing executive: " + PF);

TDS = (18 \* Salary) / 100;

Console.WriteLine("TDS of Marketing executive: " + TDS);

void CalculateSalary()

{

NetSalary = GrossSalary - (PF + TDS);

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

}

}

}

class Program

{

static void Main(string[] args)

{

Employee employee = new Employee();

Manager M = new Manager();

MarketingExecutive M1 = new MarketingExecutive();

employee.employee();

M.manager();

M.CalculateSalary();

M1.marketingExecutive();

M1.CalculateSalary();

}

}

}

4)

using System;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace file\_\_IO

{

class class1

{

public interface IEmployee

{

void ShowDetails();

}

}

public class Manager

{

private int Id;

private string Name;

private double Salary;

public Manager()

{

Console.WriteLine("Enter Id:");

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

Console.WriteLine("Enter Name:");

Name = Console.ReadLine();

Console.WriteLine("Enter Salary:");

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

}

public void ShowDetails()

{

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

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

Console.WriteLine("Salary" + Salary);

}

}

class Program

{

static void Main(string[] args)

{

void SerializeData()

{

string S = "Welcome to C#";

FileStream fileStream = new FileStream(@"C:\Users\abc\Desktop\test.txt", FileMode.Create);

BinaryFormatter formatter = new BinaryFormatter();

formatter.Serialize(fileStream, S);

fileStream.Close();

}

void DeSerializeData()

{

FileStream fileStream = new FileStream(@"C:\Users\abc\Desktop\test.txt", FileMode.Open);

BinaryFormatter binaryFormatter = new BinaryFormatter();

string data = "";

data = (string)binaryFormatter.Deserialize(fileStream);

fileStream.Close();

Console.WriteLine("your deserialize data is");

Console.WriteLine(data);

}

class1 C = new class1();

Manager manager = new Manager();

manager.ShowDetails();

SerializeData();

DeSerializeData();

Console.ReadLine();

}

}

}