|  |  |  |  |
| --- | --- | --- | --- |
|  | **COLLEGE OF COMPUTING AND INFORMATION SCIENCES** | | |
| **Final Assessment of Lab Exam (Spring 2021 Semester)** | | |
| **Class Id** | 106277 | **Course Title** | Object Oriented Programing Lab |
| **Program** | BSCS | **Campus / Shift** | Main-Morning |
| **Date** | 21-April-2021 | **Total Marks** | 20 |
| **Duration** | 2.5 hours | **Faculty Name** | Syed Affan Ahmed Zaidi |
| **Student Id** | 12067 | **Student Name** | Muhammad Areeb |
| **Code** | **OOP106277-B** |  |  |

**Instructions:**

* Fill out your Student ID and Student Name in above header.
* Do not remove or change any part question paper.
* Write down your answers with title “Answer for Question# 00”.
* Handwritten text or image should be on A4 size page with clear visibility of contents.
* In case of CHEATING, COPIED material or any unfair means would result in negative marking or ZERO.
* **Caution:** Duration to perform Final Assessment is **02 hours only and Half hour** is given to cater all kinds of odds in submission of Answer-sheet. **Therefore, if you failed to upload answer sheet on LMS (in PDF format) within 2.5 hours limit, you would be considered as ABSENT/FAILED.**

**Section A (Object Oriented Programing Concepts ) (15-Marks)**

1. Write a program to create a class employee, it consists of ID, name, department and address. All employees belong to “Computer Science” department and it can never be change by any means. Employee ID is initialized only once when Employee object is created, any further attempt to change ID should be failed. Class must have a 3 parameterized constructor to set values and two methods: **(5 Marks)**

* print(): to display all the data of a particular employee
* totalObjects(): to count and print total number of objects that has been created

In Main(), create atleast two objects of employee class, display their records by calling print() function and also print the total number of objects that has been created.

**CODE:**

using System;

namespace ConsoleApp1

{

class employee

{

public int ID;

public string Name;

private string department;

private string address;

public static int count;

public employee(int id, string name, string addres)

{

ID = id;

Name = name;

department = "Computer Science";

address = addres;

count++;

}

public void print()

{

Console.WriteLine(" ======================================= ");

Console.WriteLine("ID = " + ID);

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

Console.WriteLine("Department= " + department);

Console.WriteLine("Address= " + address);

Console.WriteLine(" ======================================= ");

}

public static void totalObjects()

{

Console.WriteLine("Total number of objects :" + count);

}

}

class Program

{

static void Main(string[] args)

{

employee obj1 = new employee(12067, "Muhammad Areeb", "Karachi");

obj1.print();

employee obj2 = new employee(12345, "Muhammad Rafiq", "Islamabad");

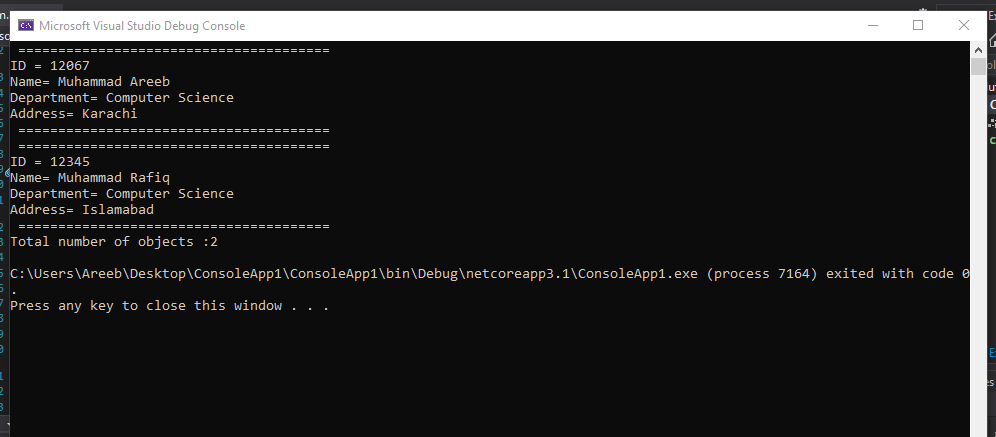
obj2.print();

employee.totalObjects();

}

}

}

**OUTPUT:**

1. There is a **book library**. Define classes respectively for a **book** and a **library**. The library must contain a name and a list of books. The books must contain the title, author, publisher, release date and ISBN-number. In the class, which describes the library, create methods to add a book to the library, to search for a book by a predefined author, to display information about a book and to delete a book from the library. Write a **test class**, which creates an object of type library, adds several books to it and displays information about each of them. Implement a test functionality, which finds all books authored by Stephen King and deletes them. Finally, display information for each of the remaining books. **(5 Marks)**

**CODE**

using System;

using System.Collections.Generic;

using System.Linq;

namespace ConsoleApp1

{

class book

{

public string Title;

public string Author;

public string Publisher;

public DateTime ReleaseDate;

public string ISBNnumber;

}

class Library

{

public string Name;

public List<book> Books = new List<book>();

public void AddBook(string title, string author, string publisher, DateTime releaseDate, string iSBNnumber)

{

book newBook = new book

{

Author = author,

Title = title,

ISBNnumber = iSBNnumber,

Publisher = publisher,

ReleaseDate = releaseDate

};

Books.Add(newBook);

}

public void DisplayBookInfo(book book)

{

Console.WriteLine($"Title-{book.Title}, Author-{book.Author}, ISBNnumber-{book.ISBNnumber}");

}

public void DeleteBook(book book)

{

Books.Remove(book);

}

public List<book> SearchBook()

{

var authorName = "Stephen King";

return Books.Where(x => x.Author == authorName).ToList();

}

}

class Test

{

static void Main(string[] args)

{

Library library = new Library();

library.AddBook("CSharp Programming", "Stephen King", "Pub1", DateTime.Now, "ISBN001");

library.AddBook("Java Programming", "J K Rowling", "Pub1", DateTime.Now, "ISBN002");

library.AddBook("Php Programming", "J K Rowling", "Pub1", DateTime.Now, "ISBN003");

library.AddBook("Ml Programming", "Stephen King", "Pub1", DateTime.Now, "ISBN004");

Console.WriteLine("Total books in Library");

if (library.Books.Count > 0)

{

foreach (var book in library.Books)

{

library.DisplayBookInfo(book);

}

Console.WriteLine("Search for all books authored by Stephen King");

var seachedBooks = library.SearchBook();

Console.WriteLine(seachedBooks.Count + " books found and deleted");

foreach (var book in seachedBooks)

{

library.DeleteBook(book);

}

Console.WriteLine("Remaining books in Library");

foreach (var book in library.Books)

{

library.DisplayBookInfo(book);

}

}

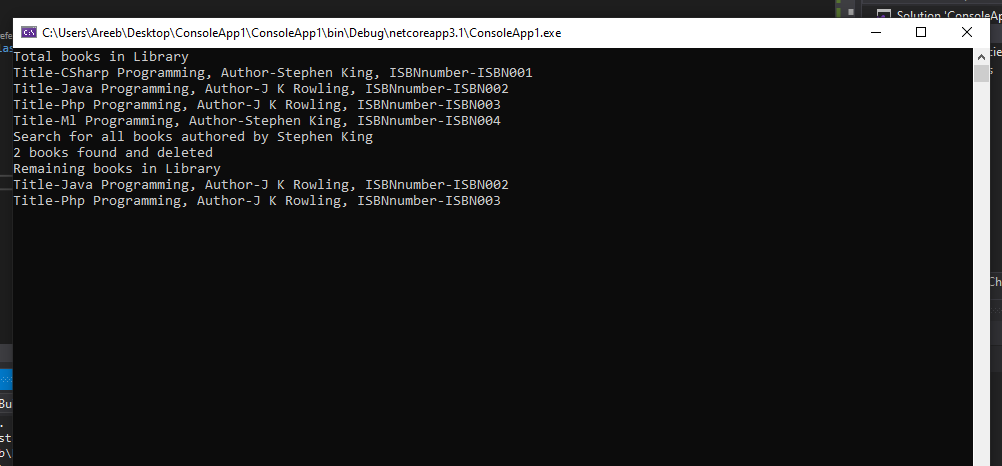
Console.ReadKey();

}

}

}

**OUTPUT:**



1. A company plans to create a **system for managing of a sound recording company**. The sound recording company has a name, address, owner and performers. Each **performer** has name, nickname and created albums. **Albums** are described with name, genre, year of creation, number of sold copies and list of songs. The **songs** are described with name and duration. Design a **set of classes with relationships** between each other, which models the data of the record company. Implement a test class, which demonstrates the work of rest of the classes. **(5 Marks)**

**CODE:**

using System;

using System.Collections.Generic;

using System.Linq;

namespace finalPaper

{

class Company

{

public string c\_name;

public string c\_address;

public string owner;

public int performers;

public Company()

{

c\_name = "COKE STUDIO ";

c\_address = "Don't let me down";

owner = "Atif Aslam";

performers = 6;

}

public void info()

{

Console.WriteLine("COMPANY DETAILS");

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

Console.Write("Company Name : " + c\_name);

Console.Write("\nCOMPANY Address: " + c\_address);

Console.Write("\nOwner : " + owner);

Console.Write("\nNo of Performers : " + performers);

}

class Performers : Company

{

public string p\_name;

public string p\_nickname;

public string p\_createdalbum;

public Performers()

{

p\_name = "Selena Gomez";

p\_nickname = "MARSHMELLO";

p\_createdalbum = "2";

}

public void info()

{

base.info();

Console.WriteLine();

Console.Write("\nPerformer Name : " + p\_name);

Console.Write("\nPerformer NickName : " + p\_nickname);

Console.Write("\nCreated Album : " + p\_createdalbum);

}

}

class Album : Performers

{

public string a\_name;

public string genre;

public string yearof\_creation;

public int num\_ofsoldcopies;

public int listof\_songs;

public Album()

{

a\_name = "JUSTICE";

genre = "SAD";

yearof\_creation = "11/10/2020";

num\_ofsoldcopies = 18;

listof\_songs = 43;

}

public void info()

{

Console.WriteLine("ALBUM DETAILS");

base.info();

Console.WriteLine();

Console.Write("\nAlbum Name : " + a\_name);

Console.Write("\nGenre : " + genre);

Console.Write("\nYear of creation: " + yearof\_creation);

Console.Write("\nNum of sold copies : " + num\_ofsoldcopies);

Console.Write("\nList of songs: " + listof\_songs);

}

}

class Song : Album

{

public string s\_name;

public int s\_duration;

public Song()

{

s\_name = "LOVE IS GONE";

s\_duration = 202;

}

public void info()

{

base.info();

Console.Write("\nSong Name : " + s\_name);

Console.Write("\nSong Duration: " + s\_duration);

}

}

class Program

{

static void Main(string[] args)

{

Song s = new Song();

s.info();

Console.ReadKey();

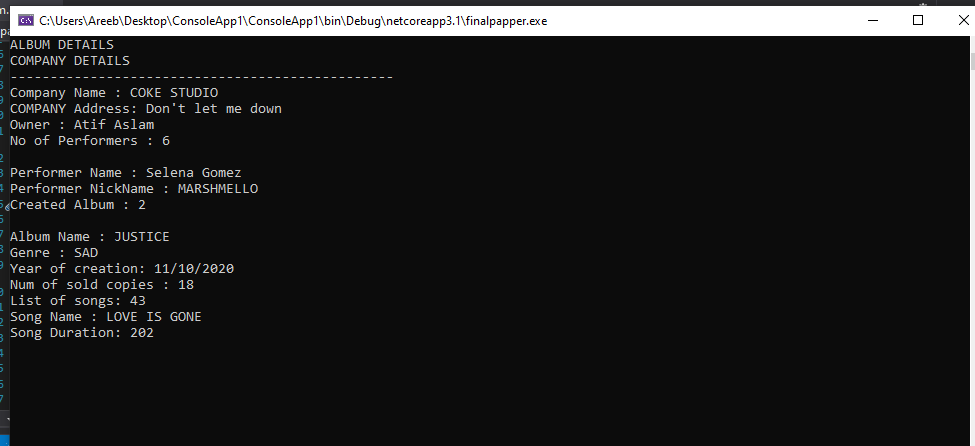
}

}

}

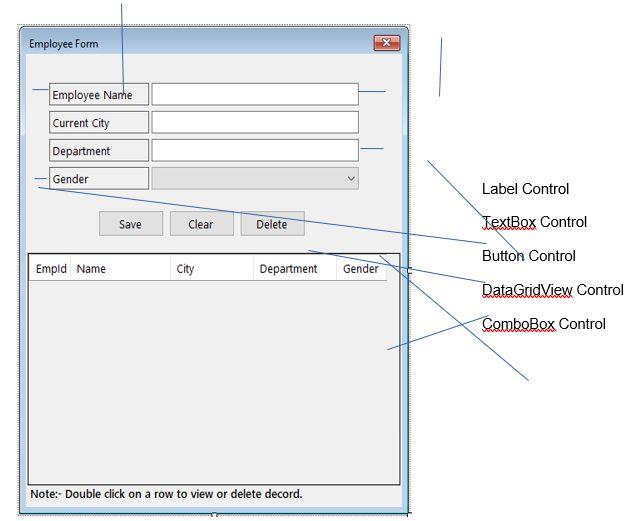
}

**OUTPUT:**



**Section B (Windows Form Application (5 Marks))**

1. Create a CRUD application which consist of following module:



**Code**

**Visual studio**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace WindowsFormsApp2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

databaseOperation con = new databaseOperation("Data Source=DESKTOP-S1PKC0F;Initial Catalog=form;Integrated Security=True");

int myid = 0;

private void Button1\_Click(object sender, EventArgs e)

{

con.insertTo("insert into info values ('" + textBox1.Text + "','" + textBox2.Text + "','" + textBox3.Text + "','" + comboBox1.Text + "')");

MessageBox.Show("Record inserted successfully", "Congrass", MessageBoxButtons.OK);

dataGridView1.DataSource = con.GetData("select \* from info");

}

private void Button3\_Click(object sender, EventArgs e)

{

if (myid > 0)

{

con.delete(@"delete from info where book\_id = '" + myid + "'");

dataGridView1.DataSource = con.GetData("select \* from info");

MessageBox.Show("Record delete successfully", "Congrass", MessageBoxButtons.OK);

clear();

}

}

public

void clear()

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

}

class databaseOperation

{

SqlConnection con;

public databaseOperation(string conString)

{

this.con = new SqlConnection(conString);

}

public DataTable GetData(string query)

{

SqlCommand cmd = new SqlCommand(query, con);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

con.Open();

adapter.Fill(dt);

con.Close();

return dt;

}

public void insertTo(string query)

{

con.Open();

SqlCommand cmd = new SqlCommand(query, con);

cmd.CommandText = query;

cmd.ExecuteNonQuery();

con.Close();

}

public void delete(string query)

{

con.Open();

SqlCommand cmd = new SqlCommand(query, con);

cmd.ExecuteNonQuery();

cmd.CommandText = query;

con.Close();

}

}

int id = 0;

private void DataGridView1\_CellMouseDoubleClick(object sender, DataGridViewCellMouseEventArgs e)

{

id = Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString());

textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();

textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();

textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();

comboBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();

}

}

}

**SQL**

create database form

use form

create table info

(

id int primary key identity(1,1),

employName varchar(50) not null,

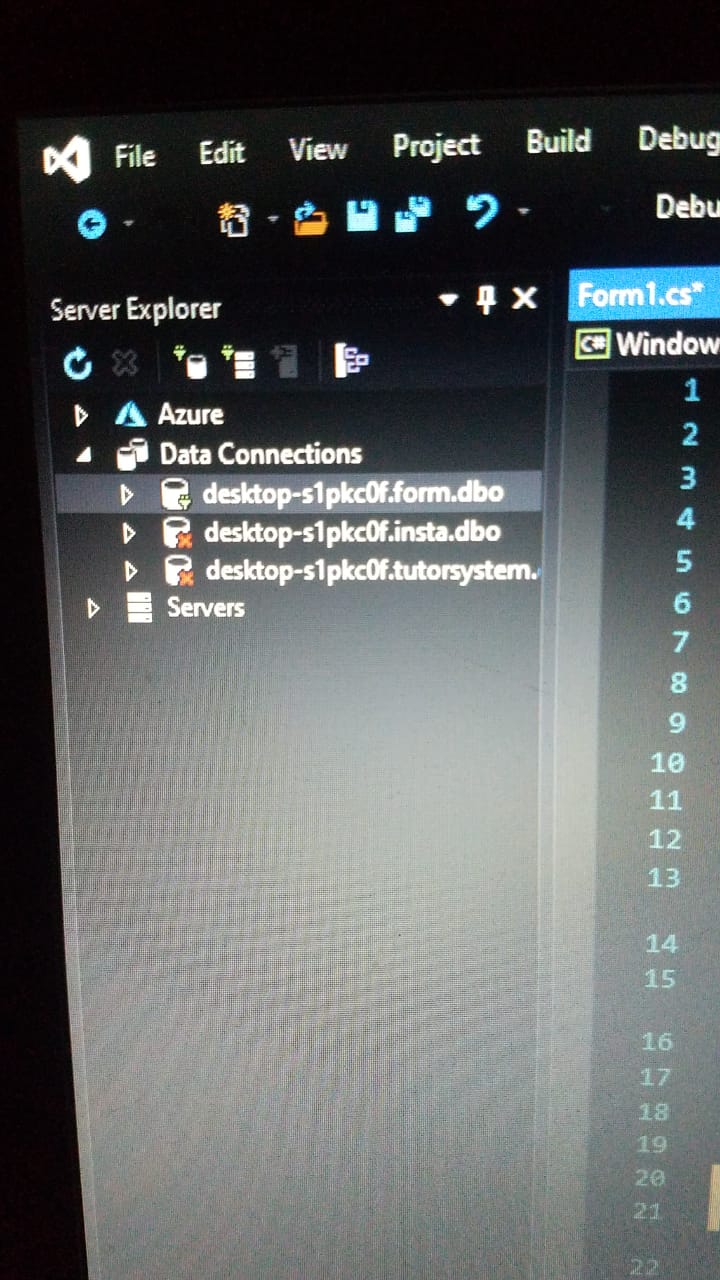
ccity varchar(50) not null,

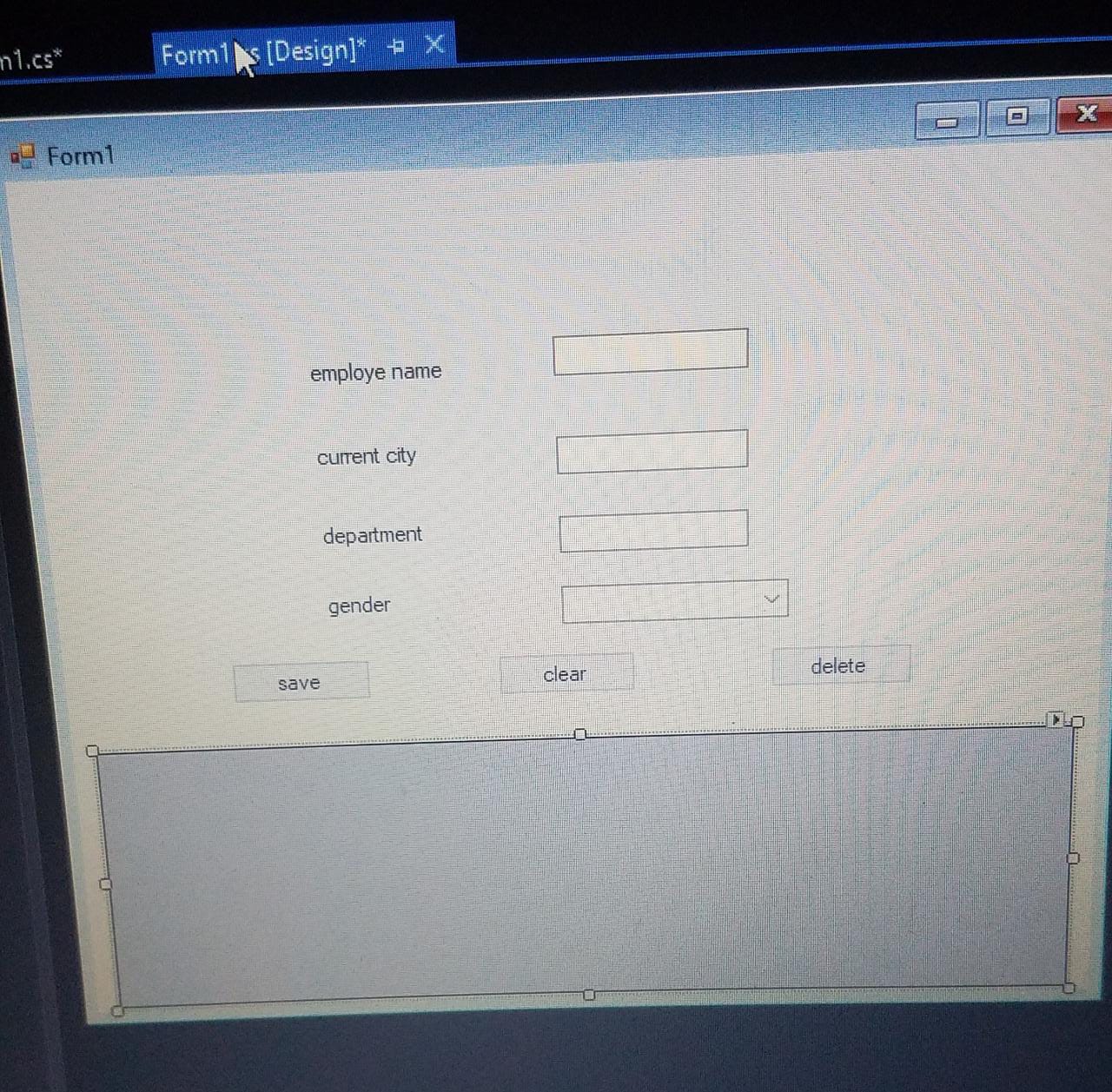
dep varchar(50) not null,

gend varchar(50) not null

)

select \* from info

**Connection**

**Form**