**Name:**

**Enrolment Number:**

|  |  |
| --- | --- |
| **Task\_1** | Rearrange the given code in “HashSetDemo.cs” file to get the expected output. |
| **ANS** | using System;  using System.Collections.Generic;    class HashSetDemo{    static public void Main()  {  HashSet<string> h1 = new HashSet<string>();  HashSet<string> h2 = new HashSet<string>();        h1.Add("C");  h1.Add("C++");  h1.Add("C#");  h1.Add("Java");  h1.Add("Ruby");  h2.Add("PHP");  h2.Add("C++");  h2.Add("Perl");  h2.Add("Java");    h1.UnionWith(h2);  foreach(var ele in h1)  {  Console.WriteLine(ele);  }  }  } |
|  |  |
|  |  |
| **Task\_2** | Write the missing statements in “MultiCastDemo.cs” file to correct the program. |
| **ANS** | using System;  delegate void DoubleOp(double x);  class MathOperations  {  public static void MultiplyByTwo(double value)  {  double result = value \* 2;  Console.WriteLine("Multiplying by 2: {0} gives {1}", value, result);  }  public static void Square(double value)  {  double result = value \* value;  Console.WriteLine("Squaring: {0} gives {1}", value, result);  }  }  class MultiCastDemo  {  static void Main(string[] args)  {  DoubleOp doubleOp = new DoubleOp(MathOperations.MultiplyByTwo);  doubleOp += MathOperations.Square;  doubleOp(2);  }  } |
|  |  |
|  |  |
| **Task\_3** | Create following Database tables.  Customer (CustID, CustName, CustCity)  Accounts(AcctID, CustID, Balance, DailyWithLimit)  Note: CustID and AcctID must be auto increment) |
| **ANS** | CREATE TABLE [dbo].[Accounts]  (  [AcctId] INT NOT NULL PRIMARY KEY IDENTITY,  [CustId] INT NULL,  [Balance] INT NULL,  [DailyWithLimit] INT NULL,  CONSTRAINT [CustId] FOREIGN KEY (CustId) REFERENCES [Customer]([CustId])  ) |
|  |  |
|  | CREATE TABLE [dbo].[Customer]  (  [CustId] INT NOT NULL PRIMARY KEY IDENTITY,  [CustName] VARCHAR(50) NULL,  [CustCity] VARCHAR(50) NULL  ) |
| **Task\_4** | Design a form and write code to insert a record into table Customer. |
| **ANS** | using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.SqlClient;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace AccountApp  {  public partial class AddCustomer : Form  {  public AddCustomer()  {  InitializeComponent();  }  private void buttonAdd\_Click(object sender, EventArgs e)  {  String name, city, conString, query;  name = textBoxName.Text;  city = textBoxCity.Text;  conString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"C:\\Users\\Sanjay Sah\\Documents\\CIE2.mdf\";Integrated Security=True;Connect Timeout=30";  SqlConnection conn = new SqlConnection(conString);  conn.Open();  query = "Insert into Customer (CustName, CustCity) Values(@name, @city)";  SqlCommand cmd = new SqlCommand(query, conn);  cmd.Parameters.AddWithValue("@name", name);  cmd.Parameters.AddWithValue("@city", city);  cmd.ExecuteNonQuery();  conn.Close();  MessageBox.Show("Added Successfully");  }  }  } |
|  |  |
|  |  |
| **Task\_5** | Design an “Accounts Form” as given below to insert, update, delete and search a record for table Accounts. (Only Designing) |
| **ANS** |  |
|  |  |
|  |  |
| **Task\_6** | Write code for insert operation in “Save” button click event.  (Refer Task\_5 Designed Form) |
| **ANS** | using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.SqlClient;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace AccountApp  {  public partial class Account : Form  {  String connString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"C:\\Users\\Sanjay Sah\\Downloads\\.net Practice\\CIE 2\\CIE2\_SuppliedFies\_Set-3\\AccountApp\\AccountApp\\CIE2.mdf\";Integrated Security=True;Connect Timeout=30";  SqlConnection conn;  SqlCommand cmd;  SqlDataReader reader;  public Account()  {  InitializeComponent();  conn = new SqlConnection(connString);  conn.Open();  bindData();  }  private void buttonSave\_Click(object sender, EventArgs e)  {  int acctId, custId, balance, dailyLimit;    custId = Convert.ToInt32(comboBoxCustId.Text);  balance = Convert.ToInt32(textBoxBalance.Text);  dailyLimit = Convert.ToInt32(textBoxDailyLimit.Text);  string query = "insert into Accounts(CustId, Balance, DailyWithLimit) values (@custId, @balance, @dailyLimit)";  cmd = new SqlCommand(query, conn);  cmd.Parameters.AddWithValue("@custId", custId);  cmd.Parameters.AddWithValue("@balance", balance);  cmd.Parameters.AddWithValue("@dailyLimit", dailyLimit);  cmd.ExecuteNonQuery();    MessageBox.Show("Inserted Successful");  }  private void bindData()  {    string query = "select CustId from Customer";  cmd = new SqlCommand(query, conn);  reader = cmd.ExecuteReader();  while (reader.Read())  {  comboBoxCustId.Items.Add(reader[0].ToString());  }  reader.Close();  }  }  } |
|  |  |
|  |  |
| **Task\_7** | Write code for delete operation in “Delete” button click event.  Write code for searching or fetching a record based on Acct\_ID in “Search” button click event.  (Refer Task\_5 Designed Form) |
| **ANS** | using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.SqlClient;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using static System.Windows.Forms.VisualStyles.VisualStyleElement;  namespace AccountApp  {  public partial class Account : Form  {  String connString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"C:\\Users\\Sanjay Sah\\Downloads\\.net Practice\\CIE 2\\CIE2\_SuppliedFies\_Set-3\\AccountApp\\AccountApp\\CIE2.mdf\";Integrated Security=True;Connect Timeout=30";  SqlConnection conn;  SqlCommand cmd;  SqlDataReader reader;  public Account()  {  InitializeComponent();  conn = new SqlConnection(connString);  conn.Open();  bindData();  }  private void buttonSave\_Click(object sender, EventArgs e)  {  int acctId, custId, balance, dailyLimit;    custId = Convert.ToInt32(comboBoxCustId.Text);  balance = Convert.ToInt32(textBoxBalance.Text);  dailyLimit = Convert.ToInt32(textBoxDailyLimit.Text);  string query = "insert into Accounts(CustId, Balance, DailyWithLimit) values (@custId, @balance, @dailyLimit)";  cmd = new SqlCommand(query, conn);  cmd.Parameters.AddWithValue("@custId", custId);  cmd.Parameters.AddWithValue("@balance", balance);  cmd.Parameters.AddWithValue("@dailyLimit", dailyLimit);  cmd.ExecuteNonQuery();    MessageBox.Show("Inserted Successful");  }  private void bindData()  {    string query = "select CustId from Customer";  cmd = new SqlCommand(query, conn);  reader = cmd.ExecuteReader();  while (reader.Read())  {  comboBoxCustId.Items.Add(reader[0].ToString());  }  reader.Close();  }  private void buttonDelete\_Click(object sender, EventArgs e)  {    string query = "delete from Accounts where AcctId = @id";  cmd = new SqlCommand(query, conn);  cmd.Parameters.AddWithValue("@id", Convert.ToInt32(textBoxAccId.Text));  cmd.ExecuteNonQuery();  MessageBox.Show("Record Deleted");  }  private void buttonSearch\_Click(object sender, EventArgs e)  {  string query = "select \* from Accounts where AcctId = @id";  cmd = new SqlCommand(query, conn);  cmd.Parameters.AddWithValue("@id", Convert.ToInt32(textBoxAccId.Text));  reader = cmd.ExecuteReader();  while (reader.Read())  {  textBoxBalance.Text = reader["Balance"].ToString();  textBoxDailyLimit.Text = reader["DailyWithLimit"].ToString();  if (reader["CustId"] == null)  {  comboBoxCustId.Text = "";  }  else  {  comboBoxCustId.Text = reader["CustId"].ToString();  }  }  reader.Close();  }  }  } |
|  |  |
|  |  |
| **Task\_8** | Write code to display records based on given Query in “View” button click event.  Query : “Get CustID, CustName, Acct\_ID, Balance, DailyWithLimit”  (Refer Task\_5 Designed Form) |
| **ANS** |  |
|  |  |
|  |  |