**Dot Net Lab Exercises**

**Introduction to Dot Net and C# basics (12)**

1. Write a program in C# to display a message as follows: “Welcome to the world of C#”
2. Write a program in C# to accept the name of the user as a command line argument and greet the user as:

“Hi! username

Welcome to the world of C#”

1. Write a program in C# to accept two numbers as command line argument and display all the numbers between the given two numbers.
2. Accept a number from the user and display whether the given number is an odd number or even number.
3. Write a program in C# to find the total number of odd and even numbers accepted from the user.
4. Write a program in C# to display temperature in Celsius. Accept the temperature in Fahrenheit.
5. A shopkeeper sells three products whose retail prices are as follows: Product 1, 22.5; product 2, 44.50; and product 3, 9.98. Write an application that reads a series of pairs of numbers as follows:

a)product number

b)quantity sold

Your application should use a switch statement to determine the retail price for each product. It should calculate and display the total retail value of all products sold. Use a loop to determine when the application should stop looping and display the final results.

1. Write a program to print the series: 0,1,4,9,16,.....625
2. Write a program in C# to find the factorial of the given number.
3. Write a program in C# to generate a Fibonacci series till 40.
4. Write a program that asks the user to type an integer N and computes the sum of the cubes from 5 to the power 3 to N the power 3
5. Write a program in C# to find the multiplication table of the given number till 20.
6. Write a program to print the numbers divisible by 7 between 200 and 300.
7. Write a program in C# to find the largest of the given three numbers. Accept the numbers from the users.
8. Write a program in C# to find the smallest of five numbers accepted from the user.
9. Write a program in C# to accept ten marks and display the following
   1. Total
   2. Average
   3. Minimum marks
   4. Maximum marks
   5. Display marks in ascending order
   6. Display marks in descending order
10. Write a program in C# to accept a word from the user and display the length of it.
11. Write a program in C# to accept a word from the user and display the reverse of it.
12. Write a program in C# to accept two words from user and find out if they are same.
13. Write a program in C# to accept a word and to find out whether the given word is a palindrome or not.

**Classes and Objects (8)**

1. Write a class Define a class to represent a bank account. Include the following members:

Data Members:

Name of the Depositor

Account Number

Type of Account

Balance amount in the account

Data Methods:

To assign initial values

To deposit an amount

To withdraw an amount after checking the balance

To display name and balance.

2. Create a class called Student with the following details:

RollNo

StudName

MarksInEng

MarksInMaths

MarksInScience

Display the total marks and Percentage of the student.

3. Change the above class definition so that you can calculate marks for five students.

4.For an Online Bookstore create a class to store book details and display the book details with fields isbn,bookname,booktitle,bookauthor,quantityofbooks,bookprice.calculate and display the bill amount

[Note: Use Properties]

5.Create a reference type called Person. Populate the Person class with the following attributes to store the following information:

First name

Last name

Email address

Date of birth

Add constructors that accept the following parameter lists:

All four parameters

First name , Last name , Email

First name , Last name , Date of birth

Write appropriate methods to accept and display the details.

6.Write an two class application that has a one-dimensional array as a data member.The array stores temperatures for any given week. Provide constructors for instantiating the class and methods to return the highest temperature, lowest temperature, average temperature, and the average temperature excluding the lowest temperature. Provide a method that accepts as an argument a temperature and returns the number of days the temperatures were below that value. Override the ToString( )

method to return the temperature range for the given week. Write a class to test your class.

7.Happy Travels is a committed tour and travel company. It has devised many innovative packages for its customers who want to take a holiday. There are three kinds of tours:

a) Discover India

b) Holiday Hungama

c) Pilgrimage Package

These tours start every week. A customer can avail of a package belonging to any category,

starting on any given date. The customer can also specify the number of people accompanying the customer.

The customer class will have the following member data:

i) customer name

ii) number of people accompanying(int)

iii) package category(D/H/P)

iv) cost(float)

v) tour start date

**Inheritance (4)**

1.A Furniture Manufacturer manufactures domestic furniture. Customers provide their specifications to the company for the furniture they want. To cope up with the received customer's order, the company decides to computerise the order-processing system. The system should accept the values of furniture items (bookshelf, table, chair, cot and dining table). You need to develop the hierarchy of these items.

[Hint : Use Abstract Class]. Write an Application that enables the company to record stock of items.

2.Imagine a publishing company that markets both book and audio-cd versions of its works. Create a class called publication that stores the title (a string) and price (type float) of a publication. From this class, derive two classes: book which adds a page count (type int); and cd, which adds a playing time in minutes. Each of these classes should have a getdata() to get its data from the user, and a putdata() method to display its data.

In the main() method write the code to test the book and tape classes by creating instances of them, asking the user to fill in the data with getdata(), and then displaying the data with putdata() through menus.

3.Create the classes required to store data regarding different types of Courses. All courses have name, duration and course fee. Courses could be part-time/full-time and courses could be onsite/online. For onsite courses, you have to store the venue name and the no. of candidates for the course. Provide constructors and the following methods. Print() and GetTotalFee()

4. Create a class to store details of student like rollno, name, course joined and fee paid so far. Assume courses are C# and ASP.NET with course fees being 2000 and 3000. Provide the a constructor to take rollno, name and course. Provide the following methods:

Payment(amount)

Print()

DueAmount property

TotalFee property

Add a static member to store Service Tax, which is set to 12.3%. Also allow a property through which we can set and get service tax.

Modify TotalFee and DueAmount properties to consider service tax

5.Create a base class, Telephone, and derive a class ElectronicPhone from it. In Telephone, create a protected string member phonetype, and a public method Ring( ) that outputs a text message like this: "Ringing the <phonetype>." In ElectronicPhone, the constructor should set the phonetype to "Digital." In the Run( ) method, call Ring( ) on the ElectronicPhone to test the inheritance.

6.Create a class called Customer with two methods

customertype() which displays the type of customer

getPriviledge() which displays the privileges according to type of the customer.

Create a class called CorporateCustomer which inherits the Customer class and overides the methods given in the Customer class.

Create a class called PersonalCustomer which inherits the Customer class and overides the methods given in the Customer class.

Create another class called MainClass containing Main method to execute the program.

7. Create a class Shape with the following methods

getDetails() to get details from the user

calculateArea() to calculate the area with the given dimensions

displayDetails() to display the calculated area of the shape.

Create a class called Triangle which inherits Shape class and provides appropriate implementation of the methods given in the base class.

Create a class called Circle which inherits Shape class and provides appropriate implementation of the methods given in the base class.

**Interface (2)**

1. Create an interface called Payable. This is the interface that will be used by the accounting department's software (which you are not responsible for authoring) for all things that they need to write checks for. The Payable interface should contain three functions:

1. Retrieve amount due

2. Add to amount due

3. Payment address

1. Derive an Employee class from the Person class already created in the previous exercise. The Employee class should add the following properties:

1. Salary

2. Mailing address

In addition, the Employee class should implement the Payable interface. The implementation of the functions specified in the Payable interface should make sense. In other words, the payment address should be the mailing address of the employee. In order to make this work right, you will need to allocate an internally protected state variable that keeps track of the amount of money due. This state variable will obviously be modified by the functions defined in the interface. You can of course, try to do this with a property and add this property to the Payable interface.

3.Write a program with abstract classes and interfaces. Abstract class should contain one abstract and non-abstract method. Abstract method should get daily sales value and return as month sales value. Non-abstract method should return daily sales value. Interface should contain one method which return sales value for a year.

Main class should inherits both class and interface. Here daily sales value is Rs.400.

System should call those three methods and display the following output:

Daily sales: Rs.400

Monthly sales: Rs.12000

Annual sales: Rs.144000

**Exception handling (2)**

**1**. Martin wants to create a ticket booking application for a movie theater. The application should ask the user for his choice, whether he wants to book the tickets or not. The application should also ask the user for the total number of tickets to be booked. While booking the tickets if the total number of booked tickets exceeds the available tickets, the application should raise an exception. Assume the total number of available tickets is 15.

**Collections & Generics (4)**

Create a string array called week which will contain the names of the days.

2. Find the length of the week array.

3. Accept 10 numbers and sort the data in ascending order and display it.

4. Write an Arraylist that will hold the names of all students and display them in descending order.

5. Write an executable program in C# that will hold the employee code and employee names available in an Organization using Collections.When the data is displayed it should be in a sorted manner.Choose an appropiate type of Collection.

6. Create a class called BookStore with fields Bookid and Bookname.Accept and display the details using HashTable.

7. Create a class with name student and store all the student details in an ArrayList and Display the Details.

**Delegates (2)**

1. Write a class Define a class to represent a bank account. Include the following members:

Data Members:

Name of the Depositor

Account Number

Type of Account

Balance amount in the account

Data Methods:

To assign initial values

To deposit an amount

To withdraw an amount after checking the balance

To display the name and the balance.

[Hint: Use delegates to call the methods]

1. Write an application that creates a class named Registration.The application would be used by the counselors of an IT education center,while registering students.The registration data entry is done differently for Career registration and for Modular registration.your application should have seperate class for the two categories of students.You need to record the student's score of the aptitude test for career registration.You need to record the student's prior experience or knowledge for modular registration. (Hint: use delegate in the registration class while registering a student to call the appropriate methods.)

**Threads (4)**

1. Write a program to display the current thread’s name.
2. Write a program to demonstrate Multithreading. The program should involve a class called Clock with two methods Tick and Tock each displaying tick and tock as messages respectively. The output should be as follows:

tick tock

tick tock

1. Write an application to simulate the vehicles crossing a toll bridge on a highway. For the purpose of this exercise, simulate the environment for five vehicles that are approaching the bridge and the toll booth. The vehicles are numbered from one to five. The vehicles should approach the bridge and the toll booth in sequential order. The toll booth can only deal with one vehicle at a time.The simulation is performed by having one thread for each vehicle and consists of three classes:

- vehicle: Simulate the behaviour of a vehicle

- TollBooth: Simulate the behaviour of a TollBooth

- Simulate : creates the vehicles and controls the simulation.

**I/O (4)**

1. Write a program to display all the drives available in the system and display (list) the contents of them.
2. Write a program to accept names of the users and store in a file.
3. Write a program to create a class called Car with Model, Year of making details. Store and retrieve this information using Binary Serialization.
4. Write a program in C# to create a log file which will hold the various errors that occurred while executing the program.

**Reflection (2)**

1. Write a program

a) to display the appdomain of the current program

b)to Create a new appdomain and run exe file generated in the previous exercise.

**Working with Visual Studio (2)**

**Simple application in VB.Net (4)**

**Working with Windows Forms (4)**

**Custom controls (2)**

**Packaging (2)**

**XML**

**Introduction to XML (4)**

1.An Online Book Store requires the details of books sold to be stored in an xml file. The book details consist of ISBN(a unique number) of the book, title of the book, the first and last names of the author of the book and the price of the book and the category of the book.

2.Create a DTD for declaring the elements to be used for storing book details in an XML document.(The categories should include Science,Fiction,Literature and Health.The ISBN should be unique).

3. Create a Schema for the XML Document containing book details.

4. Display the Book Details in a tabular format using XSLT.

**ADO.Net**

**Introduction, architecture and DataSet (4)**

1.Write a program (Console Application)

a)to Create a table called "Customers" with following Details:

Custid

Custname

CustAddress

DOB

Salary

b)Insert a few records into the table just created.

c)Retrieve(display) all the data from the table.

d)Display details based on custid supplied by the user.

e)Modify records based on the custid provided by the user.

d)Delete a record specified by the user

e)drop the table.

2.a)Create a stored procedure in SQL Server which takes a date from the user and displays all the details of the customer who are born after the given date.

b)Write a program in C# to execute the stored procedure just created.

NOTE:

Consider the following table structure of "Customers":

Custid int

Custname varchar(40)

CustAddress varchar(40)

3.Write a Windows Application program to work with Customers table using Datasets.

Perform the following operations

a)Retrieve all data

b)Retrieve data using find function

c)Insert data into the table

d)Update data for the given Custid

e)Delete record for the given Custid

4. Write an executable Windows Application program to Create A form displaying Customer Details. Use Databinding and allow navigation between the records.

**LINQ**

**Introduction (4)**

**JavaScript**

**JavaScript Language Basics (2)**

**DOM and predefined classes (4)**

**Object Library (4)**

**ASP.NET**

**ASP.NET 3.5 (4)**

**Working with Web Forms (4)**

**State Management (4)**

**Data validation (2)**

**Master Pages, skins and themes (2)**

**Developing a full-fledged application (4)**

**AJAX controls (4)**

**Website security (2)**

**Deploying ,Handling Errors and Logging (2)**

**Working with XML (4)**

ASP.NET

**Introduction to Silverlight (4)**

Web Services

**ASP.Net and Web Services (8)**

**jQuery (4)**

**Design Patterns (4)**