

TRIBHUVAN UNIVERSITY
Faculty of Humanities & Social Sciences
OFFICE OF THE DEAN

TU QUESTIONS-ANSWERS 2020

Bachelor Level / fifth-semester / Science
 Computer Science and Information Technology (CACS302)
 (DotNet Technology)
 Candidates are required to give their answer in their own words as far as practicable.

Full marks: 60
 Pass marks: 24
 Time: 3 hours

Group 'A'

Attempt all the questions.

$10 \times 1 = 10$

- i. Circle (O) the correct answer in the following questions
- i. .NET Framework is an internal component that supports the execution of applications created by using various programming languages.
 a. Internet b. Windows c. Hardware d. Language
- ii. What is CLR?
 a. Common language Runtime b. Compiler Language Runtime
 c. Compiler Library Runtime d. Common Library Runtime
- iii. Which is the following is not a component of the CLR?
 a. Security engine b. Garbage collector
 c. .NET Framework d. JIT compiler
- iv. Which of the following provides automatic memory management and resolve the issue of memory leaks and invalid memory references?
 a. Security engine b. Garbage collector
 c. JIT Compiler d. Debugger
- v. What is CTS?
 a. Common type specification b. common type safe
 c. compiler type structure d. common type system
- vi. What is CLS?
 a. compiler library specification b. common library specification
 c. compiler language specification d. common language specification
- vii. Which of the following namespace provides the classes that allow you to debug your application step by step?
 a. System b. System Diagnostics
 c. system object d. system security
- viii. What is GAC?
 a. Garbage collector b. Global assembly collector
 c. Global access cache d. Global assembly cache
- ix. What is COM?
 a. Component object model b. common object model
 c. computer oriented model d. common oriented model
- x. code that targets the common language runtime is known as....
 a. Unmanaged code b. Distributed code
 c. Managed code d. Native code

Answer-Key

- | | | | | | | | | | |
|--------|---------|----------|---------|--------|---------|----------|-----------|---------|--------|
| i. (b) | ii. (a) | iii. (c) | iv. (b) | v. (d) | vi. (d) | vii. (b) | viii. (d) | ix. (a) | x. (c) |
|--------|---------|----------|---------|--------|---------|----------|-----------|---------|--------|

Group B**Attempt any SIX questions.**

2. Differentiate Object Oriented Programming and Object Based Programming. Explain some of the major features of C# language. [6 x 5=30] [2+3]

Ans: Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods. For example, a person is an object which has certain properties such as height, gender, age, etc. It also has certain methods such as move, talk, and so on.

Object-Oriented Languages (OOP) follow all the concepts of OOPs whereas, Object-based languages don't follow all the concepts of OOPs like inheritance and polymorphism. Object-oriented languages do not have the inbuilt objects whereas Object-based languages have the inbuilt objects, for example, JavaScript has window object.

Examples for Object Oriented Languages include Java, C# whereas Object-based languages include VB etc.

Here are the significant difference between Object-oriented Programming Language and Object-based Programming Language:

Object-oriented Programming Language	Object-based Programming Language
All the characteristics and features of object-oriented programming are supported.	All characteristics and features of object-oriented programming, such as inheritance and polymorphism are not supported.
These types of programming languages don't have a built-in object. Example: C++.	These types of programming languages have built-in objects. Example: JavaScript has a window object.
Java is an example of object-oriented programming language which supports creating and inheriting (which is reusing of code) one class from another.	VB is another example of object-based language as you can create and use classes and objects, but inheriting classes is not supported.

Major features of C# language

The major features of C# from an object-oriented perspective are:

a. Unified type system

The fundamental building block in C# is an encapsulated unit of data and functions called a type. C# has a unified type system, where all types ultimately share a common base type. This means that all types, whether they represent business objects or are primitive types such as numbers, share the same basic functionality. For example, an instance of any type can be converted to a string by calling its ToString method.

b. Classes and interfaces

In a traditional object-oriented paradigm, the only kind of type is a class. In C#, there are several other kinds of types, one of which is an interface. An interface is like a class, except that it only describes members. The implementation for those members comes from types that implement the interface. Interfaces are particularly useful in scenarios where multiple inheritance is required (unlike

languages such as C++ and Eiffel, C# does not support multiple inheritance of classes).

Properties, methods, and events

- c. In the pure object-oriented paradigm, all functions are methods. In C#, methods are only one kind of function member, which also includes properties and events. Properties are function members that encapsulate a piece of an object's state, such as a button's colour or a label's text. Events are function members that simplify acting on object state changes.

While C# is primarily an object-oriented language, it also borrows from the functional programming paradigm. Specifically:

Functions can be treated as values

- a. Through the use of delegates, C# allows functions to be passed as values to and from other functions.

C# supports patterns for purity

Core to functional programming is avoiding the use of variables whose values change, in favour of declarative patterns. C# has key features to help with those patterns, including the ability to write unnamed functions on the fly that "capture" variables (lambda expressions), and the ability to perform list or reactive programming via query expressions. C# also makes it easy to define read-only fields and properties for writing immutable (read-only) types.

3. Explain overview of Microsoft .NET framework and its components in detail. [5]

Ans: DotNet is the framework for which we develop applications. It sits in between our application programs and operating system. It provides an object oriented environment. It ensures safe execution of the code by performing required runtime validations. For example, it is never possible to access an element of an array outside the boundary. Similarly, it is not possible to a program to write into another programs area, etc. The runtime validations performed by .NET makes the entire environment robust.

It is a programming infrastructure created by Microsoft for building, deploying, and running applications and services that use .NET technologies, such as desktop applications and Web services. The .NET Framework consists of:

- the Common Language Runtime
- the Framework Class Library

Common Language Runtime (CLR)

CLR is the basic and Virtual Machine component of the .NET Framework. It is the run-time environment in the .NET Framework that runs the codes and helps in making the development process easier by providing the various services such as remoting, thread management, type-safety, memory management, robustness etc. Basically, it is responsible for managing the execution of .NET programs regardless of any .NET programming language. It also helps in the management of code, as code that targets the runtime is known as the Managed Code and code doesn't target to runtime is known as Unmanaged code.

Framework Class Library (FCL)

The class library is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

It is the collection of reusable, object-oriented class libraries and methods etc. that can be integrated with CLR. Also called the Assemblies. It is just like the header files in C/C++ and packages in the java. Installing .NET framework basically is the installation of CLR and FCL into the system. The core libraries are sometimes collectively called the Base Class Library (BCL). The entire framework is called the Framework Class Library (FCL).

.Net framework provides multiple advantages to the programmers in comparison to the advantages provided by other platforms. Microsoft has integrated various modern as well as existing technologies of application development in .NET framework. These technologies are highly efficient for modern as well as future business application. There are following components of .NET framework:

1. .NET Class Library
2. Common Language runtime
3. Dynamic Language runtime
4. Application domains
5. .Net Framework Security
6. Cross Language interoperability
7. Side by side execution
8. Common Type System

1. .NET Class Library

.NET framework contains multiple classes that are readily available for developers. The classes in the FCL (framework class library) are grouped under multiple namespaces.

2. Common Language Runtime

CLR provides interoperability between different language, like C#, VB, Visual C++, by providing a common environment for the execution of code written in of these languages.

3. Dynamic Language runtime

DLR provides to execute dynamic languages on .NET Framework by adding some special services to the CLR.

4. Application domains

It is used to isolate the process of different applications and can be defined by .NET framework.

5. .NET Framework Security

.NET framework provides multiple tools that can be used by developers to protect the resources and code from unauthorized users.

6. Cross Language interoperability

Object or complied code of one language can be used in other .NET compatible language.

7. Side by side execution

In the same application we can use multiple versions of CLR simultaneously.

8. Common Type System

CTS is used to maintain data integrity across the code written in different .NET compliant languages. CTS also used to prevent data loss when a type in one language transfers data to its equivalent type in other language.

- 4.** What do you mean by property in C# language? How it is different from method? Compare automatic property with other types of property with suitable example. [1+1+3]

Ans: Property in C# is a member of a class that provides a flexible mechanism for classes to expose private fields. Internally, C# properties are special methods called accessors. A C# property have two accessors, get property accessor and set property accessor. A get accessor returns a property value, and a set accessor assigns a new value. The value keyword represents the value of a property.

Properties look like fields from the outside, but internally they contain logic, like methods do. Properties are named members of classes, structures, and interfaces. Member variables or methods in a class or structures are called Fields. Properties are an extension of fields and are accessed using the same syntax. They use accessors(get and set) through which the values of the private fields can be read, written or manipulated. Usually, inside a class, we declare a data field as private and will provide a set of public SET and GET methods to access the data fields. This is a good programming practice since the data fields are not directly accessible outside the class. We must use the set/get methods to access the data fields.

An example, which uses a set of set/get methods, is shown below.

```
using System;
class MyClass
{
    private int x;
    public void SetX(int i)
    {
        x = i
    }
    public int GetX()
    {
        return int GetX()
    }
}
class MyClient
{
    public static void Main()
    {
        MyClass mc = new MyClass();
        mc.SetX(10);
        int xVal = mc.GetX();
        ConSole.WriteLine(xVal);
    }
}
```

But C# provides a built in mechanism called properties to do the above. In C#, properties are defined using the property declaration syntax. The general form of declaring a property is as follows.

<acces_modifier><return_type><property_name>

{

get

{

}

set

{

}

}

For example, the above program can be modified with a property X as follows.

```
using System;
class MyClass
{
    private int x;
    public int X
    {
        get
        {
            return x;
        }
        set
        {
            x = value;
        }
    }
}
class MyClient
{
    public static void Main()
    {
        MyClass mc = new MyClass();
        mc.X = 10;
        int xVal = mc.X;
        Console.WriteLine(xVal); // Displays 10
    }
}
```

Compare automatic property with other types of property

Automatic Properties

The most common implementation for a property is a getter and/or setter that simply reads and writes to a private field of the same type as the property. An automatic property declaration instructs the compiler to provide this implementation. We can improve the first example in this section by declaring CurrentPrice as an automatic property:

```
public class Stock
{
```

```

    ... public decimal CurrentPrice { get; set; }
}
Example of Automatic Property
classChk
{
    public int a { get; set; }
    public int b { get; set; }
    public int sum
    {
        get
        {
            return a + b;
        }
    }
}
classTest
{
    static void Main()
    {
        Chk obj = new Chk();
        obj.a = 10;
        obj.b = 5;
        Console.WriteLine("Sum of "+obj.a+" and "+obj.b+" = "+obj.sum);
        Console.ReadKey();
    }
}

```

Output:

Sum of 10 and 5 = 15

5. Define constructor. Explain different types of constructors used in C# with example. [1+4]

Ans: A constructor in C# is a block of code similar to a method that's called when an instance of an object is created. Here are the key differences between a constructor and a method:

- A constructor doesn't have a return type.
- The name of the constructor must be the same as the name of the class.
- Unlike methods, constructors are not considered members of a class.
- A constructor is called automatically when a new instance of an object is created.

All classes have constructors, whether you define one or not, because C# automatically provides a default constructor that initializes all member variables to zero. However, once you define your own constructor, the default constructor is no longer used. In C#, constructors can be divided into 5 types

- Default Constructor
- Parameterized Constructor
- Copy Constructor

- Static Constructor
- Private Constructor

Default Constructor in C#

A constructor without any parameters is called a default constructor; in other words, this type of constructor does not take parameters. The drawback of a default constructor is that every instance of the class will be initialized to the same values and it is not possible to initialize each instance of the class with different values. The default constructor initializes:

Example

```
using System;
namespace DefaultConstructor
{
    class addition
    {
        int a, b;
        public addition() //default constructor
        {
            a = 100;
            b = 175;
        }
    }
}
```

public static void Main()

```
{  
    additionobj = new addition(); //an object is created , constructor is called  
    Console.WriteLine(obj.a);  
    Console.WriteLine(obj.b);  
    Console.Read();  
}
```

Parameterized Constructor

A constructor with at least one parameter is called a parameterized constructor. The advantage of a parameterized constructor is that you can initialize each instance of the class with a different value.

```
using System;
namespace Constructor
{
    class paraconstructor
    {
        public int a, b;
        public paraconstructor(int x, int y)
        {
            a = x;
            b = y;
        }
    }
}
```

class MainClass

```

static void Main()
{
    paraconstrctor v = new paraconstrctor(100, 175);
    Console.WriteLine("\t");
    Console.WriteLine("value of a=" + v.a );
    Console.WriteLine("value of b=" + v.b);
    Console.Read();
}
}
}
}

```

Copy Constructor

The constructor which creates an object by copying variables from another object is called a copy constructor. The purpose of a copy constructor is to initialize a new instance to the values of an existing instance.

Syntax

```

public employee(employee emp)
{
    name=emp.name;
    age=emp.age;
}

```

The copy constructor is invoked by instantiating an object of type employee and bypassing it the object to be copied. Example
employee emp1=new employee (emp2);

Static Constructor

When a constructor is created using a static keyword, it will be invoked only once for all of the instances of the class and it is invoked during the creation of the first instance of the class or the first reference to a static member in the class. A static constructor is used to initialize static fields of the class and to write the code that needs to be executed only once.

Syntax

```

class employee
{
    // Static constructor
    static employee(){}
}

```

Private Constructor

When a constructor is created with a private specifier, it is not possible for other classes to derive from this class, neither is it possible to create an instance of this class. They are usually used in classes that contain static members only. Some key points of a private constructor are:

```

using System;
namespacedefaultConstractor
{
    public class Counter
    {
        private Counter() // private constrctor declaration
    }
}

```

```

        public static int currentview;
        public static int visitedCount()
        {
            return ++ currentview;
        }
    }
    class ViewCountedDetails
    {
        static void Main()
        {
            // Counter aCounter = new Counter(); // Error
            Console.WriteLine();
            Counter.currentview = 500;
            Counter.visitedCount();
            Console.WriteLine("Now the view count is: {0}", Counter.currentview);
            Console.ReadLine();
        }
    }
}

```

6. Define inheritance. Write a C# program to demonstrate multilevel and multiple inheritance. [1+2+2]

Ans: The process by which one class acquires the properties (data members) and functionalities (methods) of another class is called **inheritance**. The aim of inheritance is to provide the reusability of code so that a class has to write only the unique features and rest of the common properties and functionalities can be extended from the another class.

The idea behind inheritance in C# is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.

The syntax of C# Inheritance

```

class Subclass-name : Superclass-name
{
    //methods and fields
}

```

The : indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.

In the terminology of C#, a class which is inherited is called a parent or superclass, and the new class is called child or subclass.

Multilevel Inheritance

Multilevel inheritance refers to a child and parent class relationship where a class extends the child class. For example, class C extends class B and class B extends class A.

Syntax of Multilevel Inheritance:

```

class A //base class
{
    //data members & methods
}

```

```

class B : A //derived class
    //also base class of C
{
    //data members & methods
}

class C : B //derived class
{
    //data members & methods
}

```

Example:

```

using System;
classA
{
    public int a, b, c;
    public void ReadData(int a, int b)
    {
        this.a = a;
        this.b = b;
    }
    public void Display()
    {
        Console.WriteLine("Value of a is: " + a);
        Console.WriteLine("Value of b is: " + b);
    }
}
classB : A
{
    public void Add()
    {
        base.c = base.a + base.b;
        Console.WriteLine("Sum=" + base.c);
    }
}
classC : B
{
    public void Sub()
    {
        base.c = base.a - base.b;
        Console.WriteLine("Difference=" + base.c);
    }
}
classLevel
{
    static void Main()
    {
        C obj = new C();
        obj.ReadData(20, 5);
    }
}

```

```

        obj.Display();
        obj.Add();
        obj.Sub();
        Console.ReadLine();
    }
}

```

Multiple Inheritance

When one class extends more than one classes then this is called multiple inheritance. For example: Class C extends class A and B then this type of inheritance is known as multiple inheritance. C# doesn't allow multiple inheritance. We can use interfaces instead of classes to achieve the same purpose.

Syntax of Multiple Inheritance:

interface A

```

{
    //methods (only signature)
}
```

class B

```

{
    //data members & methods
}
```

class C : B,A

```

{
    //data members & methods
}
```

Example:

```

using System;
interfaceA
{
    //doesn't contain data members
    //method only contain signature
    intCalculateArea();
    intCalculatePerimeter();
}
classB
{
    publicint l, b;
    publicvoidReadData(intl,int b)
    {
        this.l = l;
        this.b = b;
    }
}
classC : A, B
{
    publicintCalculateArea()
    {
        ...
    }
}

```

```

        ReadData(10,5);
        int area=l*b;
        return area;
    }

    public int CalculatePerimeter()
    {
        ReadData(15, 10);
        int peri = 2*(l+b);
        return peri;
    }

}

//driver class
class Inter
{
    static void Main(string[] args)
    {
        C obj = new C();
        Console.WriteLine("Area of Rectangle=" + obj.CalculateArea());
        Console.WriteLine("Perimeter of Rectangle=" +
        obj.CalculatePerimeter());
        Console.ReadKey();
    }
}

```

7. What is generics? List different types of generic classes. Explain delegate with example. [1+1+3]

Ans: Generics in C# and .NET procedure many of the benefits of strongly-typed collections as well as provide a higher quality of and a performance boost for code.

Generics are very similar to C++ templates but having a slight difference in such a way that the source code of C++ templates is required when a template is instantiated with a specific type and .NET Generics are not limited to classes only. In fact, they can also be implemented with Interfaces, Delegates and Methods.

Generic Classes

The Generic class can be defined by putting the <T> sign after the class name. It isn't mandatory to put the "T" word in the Generic type definition. You can use any word in the TestClass<> class declaration.

```
public class TestClass<T> {}
```

The System.Collection.Generic namespace also defines a number of classes that implement many of these key interfaces. The following table describes the core class types of this namespace.

Generic class	Description
Collection<T>	The basis for a generic collection Comparer compares two generic objects for equality
Dictionary< TKey, TValue >	A generic collection of name/value pairs
List<T>	A dynamically resizable list of Items
Queue<T>	A generic implementation of a first-in, first-out (FIFO) list
Stack<T>	A generic implementation of a last-in, first-out (LIFO) list

C# - Delegates

The delegate is a reference type data type that defines the method signature. We can define variables of delegate, just like other data type, that can refer to any method with the same signature as the delegate. There are three steps involved while working with delegates:

- Declare a delegate
- Set a target method
- Invoke a delegate

A delegate can be declared using the delegate keyword followed by a function signature, as shown below.

Delegate Syntax

[Access modifier] delegate [return type] [delegate name]([parameters])

Complete Example:

```
using System;
public delegate void MyDelegate(string msg);
public class Program
{
    public static void Main()
    {
        MyDelegatedel = ClassA.MethodA;
        del("Hello World");
        del = ClassB.MethodB;
        del("Hello World");
        del = (string msg) => Console.WriteLine("Called lambda
expression: " + msg);
        del("Hello World");
    }
}
public class ClassA
{
    public static void MethodA(string message)
    {
        Console.WriteLine("Called ClassA.MethodA() with parameter: "+ message);
    }
}
public class ClassB
{
    public static void MethodB(string message)
    {
        Console.WriteLine("Called ClassB.MethodB() with parameter: "+ message);
    }
}
```

Input / Output

Called ClassA.MethodA() with parameter: Hello World

Called ClassB.MethodB() with parameter: Hello World

Called lambda expression: Hello World

8. What do you mean by lambda expression? Explain different types of lambda expression used in C# with example. [1+4]

Ans: Lambda expressions in C# are used like anonymous functions, with the difference that in Lambda expressions you don't need to specify the type of

the value that you input thus making it more flexible to use. The ' $=>$ ' is the lambda operator which is used in all lambda expressions. The Lambda expression is divided into two parts, the left side is the input and the right is the expression.

The Lambda Expressions can be of two types:-

1. Expression Lambda: Consists of the input and the expression.

Syntax:

 input $=>$ expression;

2. Statement Lambda: Consists of the input and a set of statements to be executed. It can be used along with delegates.

Syntax:

 input $=>\{$ statements $\}$;

Basic example of lambda expression:

```
classLambdaTest
{
    static int test1() => 5;
    static int test2(int x) => x + 10;
    static void Main(string[] args)
    {
        int x = test1();
        int res = test2(x);
        Console.WriteLine("Result is: " + res);
    }
}
```

Group C

Attempt any TWO questions.

[2x10=20]

9. a) Write a program to create user registration form in one ASP.NET web page and display filled data in another page. [7]

Ans:

```
<body>
<form id="form">
<div>
<table class="auto-style1">
<tr>
<td>
<asp:Label ID="Label1" Text="User Name"></asp:Label>
</td>
<td>
<asp:TextBox ID="username" required="true"></asp:TextBox></td>
</tr>
<tr>
<td>
<asp:Label ID="Label6" Text="Email ID"></asp:Label>
</td>
<td>
<asp:TextBox ID="EmailID" TextMode="Email"></asp:TextBox></td>
</tr>
<tr>
<td>
<asp:Label ID="Label2" Text="Password"></asp:Label></td>
```

```

<td>
<asp:TextBoxID="TextBox2"
TextMode="Password"></asp:TextBox></td>
</tr>
<tr>
<td>
<asp:Label ID="Label3" Text="Confirm Password"></asp:Label></td>
<td>
<asp:TextBoxID="TextBox3" TextMode="Password"></asp:TextBox></td>
</tr>
<tr>
<td>
<asp:Label ID="Label4" Text="Gender"></asp:Label></td>
<td>
<asp:RadioButton ID="RadioButton1" GroupName="gender" Text="Male" />
<asp:RadioButton ID="RadioButton2" GroupName="gender" Text="Female" /></td>
</tr>
<tr>
<td>
<asp:Label ID="Label5" Text="Select Course"></asp:Label>s</td>
<td>
<asp:CheckBox ID="CheckBox1" />
<asp:CheckBox ID="CheckBox2" />
<asp:CheckBox ID="CheckBox3" />
</td>
</tr>
<tr>
<td>
</td>
<td>
<br />
<asp:Button ID="Button1" Text="Register" CssClass="btn btn-primary"
OnClick="Button1_Click"/>
</td>
</tr>
</table>
<asp:Label ID="message" Font-Size="Medium"
ForeColor="Red"></asp:Label>
</div>
</form>
<table class="auto-style1">
<tr>
<td class="auto-style2"><asp:Label ID="ShowUserNameLabel" /></asp:Label></td>
<td>
<asp:Label ID="ShowUserName" ></asp:Label></td>
</tr>
<tr>

```

```

<td class="auto-style2"><asp:Label ID="ShowEmailIDLabel" >
</asp:Label></td>
<td>
<asp:Label ID="ShowEmail" ></asp:Label></td>
</tr>
<tr>
<td class="auto-style3"><asp:Label ID="ShowGenderLabel" >
</asp:Label></td>
<td class="auto-style4">
<asp:Label ID="ShowGender" ></asp:Label></td>
</tr>
<tr>
<td class="auto-style2"><asp:Label ID="ShowCourseLabel" >
</asp:Label></td>
<td>
<asp:Label ID="ShowCourses" ></asp:Label></td>
</tr>
</table>
</body>
</html>

```

b) Write a program for handling exception in ASP.NET. [3]

Ans: An exception is a problem that arises during the execution of a program. A C# exception is a response to an exceptional circumstance that arises while a program is running, such as an attempt to divide by zero.

Exceptions provide a way to transfer control from one part of a program to another. C# exception handling is built upon four keywords: **try, catch, finally, and throw**.

- **try** – A try block identifies a block of code for which particular exceptions are activated. It is followed by one or more catch blocks.
- **catch** – A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The catch keyword indicates the catching of an exception.
- **finally** – The finally block is used to execute a given set of statements, whether an exception is thrown or not thrown. For example, if you open a file, it must be closed whether an exception is raised or not.
- **throw** – A program throws an exception when a problem shows up. This is done using a throw keyword.

Example:

```

using System;
class HandledException
{
    public static void Main()
    {
        int x = 0;
        int intTemp = 0;
        try
        {
            intTemp = 100/x;
        }
    }
}

```

```

        Console.WriteLine("Not executed line");
    }
catch
{
    Console.WriteLine("oException" );
}
Console.WriteLine("Result is {0}", intTemp);
}
}
}

```

10. a) How virtual method is used to achieve dynamic binding in C#? Explain with the help of suitable program. [1+4]

Ans: Polymorphism, Late Binding, Method overriding etc. Whereas in static polymorphism we overload a function; in dynamic polymorphism we override a base class function using virtual or override keyword. Method overriding means having two or more methods with the same name, same signature but with different implementation.

Example

```

using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
namespaceRuntimePolymorphism
{
    class Program
    {
        static void Main(string[] args)
        {
            Chocolate ch = new Chocolate();
            ch.flavor();
            Console.ReadKey();
        }
    }
    classIceCream
    {
        publicIceCream()
        {
            Console.WriteLine("Class : Icecream");
        }
        public virtual void flavor()
        {
            Console.WriteLine("IceCream Type : Vanilla");
        }
    }
    class Chocolate : IceCream
    {
        public Chocolate()
        {
            Console.WriteLine("Class : Chocolate");
        }
    }
}

```

```

        }
    public override void flavor()
    {
        Console.WriteLine("IceCream Type : Chocolate");
    }
}

```

- b) Define operator overloading. Write a C# program to overload binary operator. [1+4]

Ans: Operator overloading gives the ability to use the same operator to do various operations. It enables to make user-defined implementations of various operations where one or both of the operands are of a user-defined class. Only the predefined set of C# operators can be overloaded. To make operations on a user-defined data type is not as simple as the operations on a built-in data type. To use operators with user-defined data types, they need to be overloaded according to a programmer's requirement. An operator can be overloaded by defining a function to it. The function of the operator is declared by using the operator keyword.

Syntax:

```

accessspecifierclassName operatoroperator_symbol
(parameters)
{
    // Code
}

```

C# program to illustrate the Binary Operator Overloading

```

using System;
namespaceBinaryOverload
{
    class Calculator
    {
        public int number = 0;
        // no-argument constructor
        public Calculator() {}
        // parameterized constructor
        public Calculator(int n)
        {
            number = n;
        }
        // Overloading of Binary "+" operator
        public static Calculator operator + (Calculator Calc1, Calculator Calc2)
        {
            Calculator Calc3 = new Calculator(0);
            Calc3.number = Calc2.number + Calc1.number;
            return Calc3;
        }
        // function to display result
        public void display()
        {
            Console.WriteLine("{0}", number);
        }
    }
}

```

```

        }
    }

classCalNum
{
    // Driver Code
    static void Main(string[] args)
    {
        Calculator num1 = new Calculator(200);
        Calculator num2 = new Calculator(40);
        Calculator num3 = new Calculator();
        num3 = num1 + num2;
        num1.display(); // Displays 200
        num2.display(); // Displays 40
        num3.display(); // Displays 240
    }
}
}

```

11. a) What is LINQ? Write a program to select employees whose salary is greater than 20000 and whose address is Kathmandu using LINQ. [1+4]

Ans: LINQ (Language Integrated Query) is uniform query syntax in C# to retrieve data from different sources and formats. It is integrated in C#, thereby eliminating the mismatch between programming languages and databases, as well as providing a single querying interface for different types of data sources.

For example, SQL is a Structured Query Language used to save and retrieve data from a database. In the same way, LINQ is a structured query syntax built in C# to retrieve data from different types of data sources such as collections, ADO.Net DataSet, XML Docs, web service and MS SQL Server and other databases.

Program part,

```

using System;
using System.Collections.Generic;
using System.Linq;
public class Employee
{
    int ID;
    string Name;
    string Address;
    int Salary;
    public override string ToString()
    {
        return ID + " " + Name + " " + Address + " " + Salary;
    }
    static void Main(string[] args)
    {
        List<Employee> employees = new List<Employee>()
        {
            new Employee {ID=101, Name="Sumit" ,Address='Palpa', Salary=4000},

```

```

new Employee {ID=102,      Name="Kiran"      ,Address='Kathmandu',
Salary=6000},
new Employee {ID=103,      Name="Suman"      , Address ='Kanchanpur',
Salary=7000},
new Employee {ID=104,      Name="Raman"      , Address ='Kathmandu',
Salary=90000},
};

IEnumerable<Employee> Query =
fromemp in employees
where emp.Salary > 20000 AND Address='Kathmandu'
selectemp;
Console.WriteLine("ID Name Address Salary");
Console.WriteLine("=====");
foreach (Employee s in Query)
{
    Console.WriteLine(s.ToString());
}
Console.WriteLine("=====");
}
}

```

- b) Write a C# program to show insert and select operation in database. [5]

Ans:

```

using MySql.Data.MySqlClient;
using System;
using System.Data;
namespace DatabaseTest
{
    classProgram
    {
        MySqlConnection conn;
        MySqlCommand command;
        voidCreateConnection()
        {
            stringconstr = "SERVER=localhost;
DATABASE=dbtest;
UID=root; PASSWORD=;";
            conn = new MySqlConnection(constr);
            conn.Open();
        }
        voidSelectRecords(stringsql)
        {
            command = new MySqlCommand(sql, conn);
            MySqlDataAdapter adapter =
            new MySqlDataAdapter(command);
            DataTabledt = new DataTable();
            adapter.Fill(dt);
            if (dt.Rows.Count != 0)
            {
                Console.WriteLine("Sid\tName\t
Address");
            }
        }
    }
}

```

```

        for (int i = 0; i < dt.Rows.Count; i++)
    {
        string sid =
            dt.Rows[i]["sid"].ToString();
        string name =
            dt.Rows[i]["name"].ToString();
        string address =
            dt.Rows[i]["address"].ToString();
        Console.WriteLine(sid + "\t" + name
            + "\t" + address);
    }
}

static void Main(string[] args)
{
    Program obj = new Program();
    try
    {
        obj.CreateConnection();
        x: Console.WriteLine("1. Insert \t 2. Select");
        Console.WriteLine("Enter your choice: ");
        int n = Convert.ToInt32(Console.ReadLine());
        string sql = "", nm = "", add = "";
        int id = 0;
        switch (n)
        {
            case 1:
                Console.WriteLine("Enter Name of Student: ");
                nm = Console.ReadLine();
                Console.WriteLine("Enter Address of Student: ");
                add = Console.ReadLine();
                sql = "INSERT INTO
tblStudent(name,address)VALUES('" + nm + "','" + add + "')";
                obj.InsertUpdateDelete(sql);
                break;
            case 2:
                sql = "SELECT * FROM tblStudent";
                obj.SelectRecords(sql);
                break;
            default:
                Console.WriteLine("Wrong Choice");
                break;
        }
        goto x;
    }
}

```



MODEL QUESTIONS SETS FOR PRACTICE

MODEL SET 1

Bachelor Level / fifth-semester / Science
 Computer Science and Information Technology (CACS302)
 (DotNet Technology)

Candidates are required to give their answer in their own words as far as practicable.

Full marks: 60

Pass marks: 24

Time: 3 hours

Group 'A'

Attempt all the questions.

$10 \times 1 = 10$

1. Circle (O) the correct answer in the following questions

- i. What is CLS?
 a. Compiler Library Specification b. Common Library Specification
 c. Compiler Language Specification d. Common Language Specification
- ii. What is DLL?
 a. Dynamic-Link Language b. Direct Link Language
 c. Dynamic-Link Library d. Direct Link Library
- iii. Which Of The Following Namespace Provides The Classes That Allow You To Debug Your Application Step By Step?
 a. System b. System .Diagnostics
 c. System.Object d. System.Security
- iv. What Is GAC?
 a. Garbage collector b. Global assembly collector
 c. Global access cache d. Global assembly cache
- v. What Is COM?
 a. Component object model b. Common object model
 c. Computer oriented model d. Common oriented model
- vi. Which of the following components of the .NET framework provide an extensible set of classes that can be used by any .NET compliant programming language?
 a. .NET class libraries b. Common Language Runtime
 c. Common Language Infrastructure d. Component Object Model
- vii. Which of the following statements correctly define .NET Framework?
 a. It is an environment for developing, building, deploying and executing Desktop Applications, Web Applications and Web Services.
 b. It is an environment for developing, building, deploying and executing only Web Applications.
 c. It is an environment for developing, building, deploying and executing Distributed Applications.
 d. It is an environment for developing, building, deploying and executing Web Services.
- viii. Which of the following constitutes the .NET Framework?
 a. ASP.NET Applications b. CLR
 c. Framework Class Library d. None of above

- ix. Code that targets the Common Language Runtime is known as
 - a. Unmanaged code
 - b. Distributed code
 - c. Managed Code
 - d. Native Code
- x. Which of the following statements is correct about Managed Code?
 - a. Managed code is the code that is compiled by the JIT compilers.
 - b. Managed code is the code where resources are Garbage Collected.
 - c. Managed code is the code that runs on top of Windows.
 - d. Managed code is the code that is written to target the services of the CLR.

Group B

Attempt any SIX questions.

[6x5=30]

- 1. What are different LINQ operators? Explain.
- 2. Explain indexers with the help of suitable example.
- 3. What do you mean by static class? Explain with suitable program.
- 4. Write a C# program to find product of any two matrices.
- 5. What are other popular frameworks related to .NET? Explain.
- 6. Explain different .NET implementations in brief.
- 7. Differentiate OOP and POP in detail.

Group C

Attempt any TWO questions.

[2x10=20]

- 1. (a) Differentiate single and multilevel inheritance with example.
(b) Differentiate error and exception with example. Write a suitable program to demonstrate exception handling.
- 2. (a) Write a program to show the use of auto-increment and auto-decrement operators.
(b) Explain DataReader, DataAdapter, Dataset and Datatable with example.
- 3. (a) What do you mean by generics? Write a C# program to create generic class and generic methods.
(b) Write a C# program to find factorial of any number.

MODEL SET 2

Group A

Attempt all the questions

[10x1=10]

- 1. Circle (O) the correct answer in the following questions:
 - i. Code snippet for having a named instance of SQL Server would be.....
 - a) "Server=localhost\sqlexpress"
 - b) "Server=local\sqlexpress"
 - c) "Server=host\sqlexpress"
 - d) "Ser=localhost\sqlexpress"
 - ii. Syntax for closing and opening the connection in ADO.net is.....
 - a) sqlConn.Open() and sqlConn.close()
 - b) sqlConn.open() and sqlConn.Close()
 - c) sqlConn.Open() and sqlConn.Close()
 - d) none of the mentioned
 - iii. _____ object is used to fill a DataSet/DataTable with query results in ADO.net.
 - a) DataReader
 - b) Dataset
 - c) DataAdapter
 - d) DataTables

- iv. ASP stands for.....
 a) Active Server Pages b) Access Server Pages
 c) Active Server Platform d) Active Server Programming
 Web.config file is used.....
- v. a) Configures the time that the server-side codebehind module is called.
 b) To store the global information and variable definitions for the application
 c) To configure the web server
 d) To configure the web browser
- vi. Difference between Response.Write() and Response.Output.Write().
 a) Response.Output.Write() allows you to buffer output
 b) Response.Output.Write() allows you to write formatted output
 c) Response.Output.Write() allows you to flush output
 d) Response.Output.Write() allows you to stream output
- vii. What class does the ASP.NET Web Form class inherit from by default?
 a) System.Web.UI.Page b) System.Web.UI.Form
 c) System.Web.GUI.Page d) System.Web.Form
- viii. Attribute must be set on a validator control for the validation to work.....
 a) ControlToValidate b) ControlToBind
 c) ValidateControl d) Validate
- ix. The DataReader object is an alternative to the.....
 a) DataSet b) DataAdapter
 c) Both A and B d) None of the above
- x. The file extension an ASP.NET web form is.....
 a) .docx b) .aspx
 c) .jpeg d) none of the above

Group B**Attempt any SIX questions.****[6x5=30]**

1. Write a suitable program to demonstrate LINQ query.
2. What do you mean by properties in C#? Explain with example.
3. Explain automatic property in C# with example.
4. Write a C# program to add any two matrices.
5. Explain different backend technologies associated with .NET framework.
6. Explain the scope of .NET technology.
7. What is OOP? Explain features of OOP.

Group C**Attempt any TWO questions.****[2x10=20]**

1. (a) Create student registration form in ASP.NET with proper form validations.
(b) What are different validation controls used in ASP.NET? Explain with example.
2. Write a program for entering college general information in one ASP page and display it in another page.
3. (a) Differentiate class and struct. Write a C# program using struct.
(b) Explain different access supported by C# with example.


MODEL SET 3

Candidates are required to give their answer in their own words as far as practicable.
Group-A

Attempt all the questions

1. Circle (O) the correct answer in the following questions: [10x1=10]

- i. Correct method to define + operator is?
 - a) public sample operator +(int a, int b)
 - b) public abstract operator +(int a, int b)
 - c) public static sample operator +(int a, int b)
 - d) public abstract sample operator +(int a, int b)
- ii. Which keyword is used to declare a base class method while performing overriding of base class methods?
 - a) this
 - b) virtual
 - c) Override
 - d) extend
- iii. The process of defining a method in a subclass having same name & type signature as a method in its superclass is known as?
 - a) Method overloading
 - b) Method overriding
 - c) Method hiding
 - d) None of the mentioned
- iv. Which of the given modifiers can be used to prevent Method overriding?
 - a) Static
 - b) Constant
 - c) Sealed
 - d) final
- v. Select the correct statement from the following?
 - a) Static methods can be a virtual method
 - b) Abstract methods can be a virtual method
 - c) When overriding a method, the names and type signatures of the override method must be the same as the virtual method that is being overridden
 - d) We can override virtual as well as no virtual methods
- vi. Which of the following cannot be used to declare a class as a virtual?
 - a) Methods
 - b) Properties
 - c) Events
 - d) Fields
- vii. Where the properties can be declared?
 - a) Class
 - b) struct
 - c) Interface
 - d) All of the mentioned
- viii. Select the modifiers which can be used with the properties?
 - a) Private
 - b) Public
 - c) Protected Internal
 - d) All of the mentioned
- ix. Select the correct statement about properties of read and write in C#.NET?
 - a) A property can simultaneously be read or write only
 - b) A property cannot be either read only or write only
 - c) A write only property will only have get accessor
 - d) A read only property will only have get accessor
- x. What is meant by the term generics?
 - a) parameterized types
 - b) class
 - c) structure
 - d) interface

Group B

[6x5=30]

Attempt any SIX questions.

1. What are different exception classes? Explain.
2. What do you mean by LINQ? Differentiate LINQ and SQL with example.
3. What do you mean by static constructor? How it is different than other types of constructors? Explain with example.

4. Explain the use of this keyword in C# with example.
5. Write a C# program to sort name of 10 persons in alphabetical order.
6. Differentiate parameters pass by value and by reference with example.
7. Explain operator precedence with example.

Group C**Attempt any TWO questions.****[2x10=20]**

1. Write a program ASP.NET to find simple interest. Use text fields for inputs and label for output. Your program should display the result when the user presses a button.
2. (a) what is polymorphism? Explain compile time and runtime polymorphism with program in detail.
(b) Write a program to show the use of break, continue and return.
3. Write a program using ADO.NET to extract name of those students who lives in Palpa district, assuming that the student table has four attributes (ID, name, district, and age).

MODEL SET 4**Candidates are required to give their answer in their own words as far as practicable.****Group-A****Attempt all the questions****[10 x1=10]**

1. Circle (O) the correct answer in the following questions:
 - i. Which of the following is the correct way of implementing an interface addition by class maths?
 - a) classmaths : addition {}
 - b) class maths implements addition {}
 - c) class maths imports addition {}
 - d) none of the mentioned
 - ii. Access specifiers which can be used for an interface are?
 - a) Public
 - b) Protected
 - c) Private
 - d) All of the mentioned
 - iii. The number of levels of inheritance are?
 - a) 5
 - b) 4
 - c) 3
 - d) 2
 - iv. Select the class visibility modifiers among the following:
 - a) Private, protected, public, internal
 - b) Private, protected, public, internal, protected internal
 - c) Private, protected, public
 - d) All of the mentioned
 - v. In Inheritance concept, which of the following members of base class are accessible to derived class members?
 - a) static
 - b) protected
 - c) private
 - d) shared
 - vi. A class member declared protected becomes member of subclass of which type?
 - a) public member
 - b) private member
 - c) protected member
 - d) static member
 - vii. Which form of inheritance is not supported directly by C# .NET?
 - a) Multiple inheritance
 - b) Multilevel inheritance
 - c) Single inheritance
 - d) Hierarchical inheritance
 - viii. If no access modifier for a class is specified, then class accessibility is defined as?
 - a) public
 - b) protected
 - c) Private
 - d) internal

- ix. The process of defining two or more methods within the same class that have same name but different parameters list?
- Method overloading
 - Method overriding
 - Encapsulation
 - None of the mentioned
- x. Which of these can be overloaded?
- Constructors
 - Methods
 - Both Constructors & Methods
 - None of the mentioned

Group B

[6x5=30]

Attempt any SIX questions.

- What are different validation controls used in ASP.NET? Explain with example.
- Explain exception handling in ASP page with suitable program.
- Write a C# program to demonstrate group by operation in LINQ.
- Differentiate constructor and destructor with example.
- What is namespace? Explain the use of namespace in C# program with example.
- What are other popular frameworks related to .NET? Explain.
- Explain different .NET implementations in brief.

Group C

[2x10=20]

Attempt any TWO questions.

- Write a program ASP.NET to find simple interest. Use text fields for inputs and label for output. Your program should display the result when the user presses a button.
- Write a program using ADO.NET to extract name and age of those students who lives in Banepa city, assuming that the student table has four attributes (ID, name, city, and age).
- (a) What do you mean by operators? Explain different operators used in C# with examples.
 (b) What is multiple inheritance? How it is achieved in C#? Explain with suitable example.

MODEL SET 5

Candidates are required to give their answer in their own words as far as practicable.

Group-A

Attempt all the questions

[10 x1=10]

- Circle (O) the correct answer in the following questions:
 - The modifier used to define a class which does not have objects of its own but acts as a base class for its subclass is?
 - Sealed
 - Static
 - New
 - Abstract
 - Choose the correct statements among the following:
 - An abstract method does not have implementation
 - An abstract method can take either static or virtual modifiers
 - An abstract method can be declared only in abstract class
 - All of the mentioned
 - Which of the following modifiers is used when an abstract method is redefined by a derived class?
 - Overloads
 - Override
 - Base
 - Virtual

- iv. Which among the following cannot be used as a datatype for an enum in C#.NET?

 - a) short
 - b) double
 - c) int
 - d) all of the mentioned

v. Which operator among the following signifies the destructor operator?

 - a) ::
 - b) :
 - c) ~
 - d) &

vi. Operator used to free the memory when memory is allocated?

 - a) new
 - b) free
 - c) delete
 - d) none of the mentioned

vii. Select wrong statement about destructor in C#?

 - a) A class can have one destructor only
 - b) Destructors cannot be inherited or overloaded
 - c) Destructors can have modifiers or parameters
 - d) All of the mentioned

viii. What is the return type of destructor?

 - a) int
 - b) void
 - c) float
 - d) none of the mentioned

ix. How many values does a function return?

 - a) 0
 - b) 2
 - c) 1
 - d) any number of values

ix. The capability of an object in Csharp to take number of different forms and hence display behaviour as according is known as _____

 - a) Encapsulation
 - b) Polymorphism
 - c) Abstraction
 - d) None of the mentioned

Group B

Attempt any SIX questions.

$$[6 \times 5 = 30]$$

1. Explain any five form control elements with example.
 2. Create student registration form in ASP.NET with proper form validations.
 3. Write a C# program to demonstrate order by operation in LINQ.
 4. Write a C# program to find maximum and minimum number from array.
 5. What is ASP.NET? Explain different elements of ASP.NET with example.
 6. Explain any three operators with example.
 7. Write a C# program to generate multiplication table from 1 to 10.

Group C

Attempt any TWO questions.

$$[2 \times 10 = 20]$$

1. Write a program for entering college general information in one ASP page and display it in another page.

2. (a) Write a C# program to sort 'n' numbers in ascending order.
(b) What do you mean by .NET framework? Explain the overview of .NET framework in detail.

3. (a) What is LINQ? Write a program to select employees whose salary is greater than 10000 and whose address is Palpa using LINQ.
(b) Write a C# program to show update and delete operation in database.

MODEL SET 6

Candidates are required to give their answer in their own words as far as practicable.
Group—A

Attempt all the questions

1. Circle (O) the correct answer in the following questions:

[10 x 1 = 10]

- i. Which of the following constitutes the .NET Framework?
 - a. ASP.NET Applications
 - b. CLR
 - c. Framework Class Library
 - d. None of above
- ii. Code that targets the Common Language Runtime is known as
 - a. Unmanaged code
 - b. Distributed code
 - c. Managed Code
 - d. Native Code
- iii. What is the return type of constructors?
 - a) int
 - b) float
 - c) void
 - d) none of the mentioned
- iv. Which method has the same name as that of its class?
 - a) delete
 - b) class
 - c) constructor
 - d) none of the mentioned
- v. In a connection string _____ represents name of the database
 - a) Data Source
 - b) Initial Catalog
 - c) Catalog Initial
 - d) Database
- vi. What are the Command Object Methods?
 - a) ExecuteNonQuery
 - b) ExecuteReader
 - c) ExecuteScalar
 - d) All of the above.
- vii. Which ado.net class provide disconnected environment?
 - a) DataReader
 - b) DataSet
 - c) Command
 - d) None of the above.
- viii. Which database is the ADO.NET SqlConnection object designed for?
 - a) Access
 - b) Microsoft SQL Server
 - c) MySQL
 - d) Oracle
- ix. The first step of configuring a DataAdapter is to select:
 - a) An adapter object.
 - b) A connection object.
 - c) A database object.
 - d) A dataset object.
- x. To check the status of the connection, _____ property is useful
 - a) ConnectionStatus
 - b) Status
 - c) State
 - d) ConnectionState

Group B

Attempt any SIX questions.

[6x5=30]

1. Write a C# program to demonstrate complete CRUD operation of a Student.
2. What is ASP.NET? Explain different elements of ASP.NET with example.
3. Write a C# program to demonstrate aggregate functions in LINQ.
4. Define constructor. Explain parameterized constructors used in C# with example.
5. Define inheritance. Write C# program to demonstrate hybrid inheritance.
6. What do you mean by method overriding? Explain the use of virtual function for method overriding.

7. What is operator overloading? What are different operators that can be overloaded in C#? Explain.

Group C**Attempt any TWO questions.****[2x10=20]**

1. (a) Write a C# program to sort given elements in descending order.
(b) What is polymorphism? Explain compile time and runtime polymorphism with program in detail.
2. (a) Differentiate ADO and ADO.NET with example.
(b) Explain the process of creating connection in ADO.NET.
3. (a) What is generics? List different types of generic classes. Explain delegate with example.
(b) What do you mean by lambda expression? Explain different types of lambda expression used in C# with example.

MODEL SET 7

Candidates are required to give their answer in their own words as far as practicable.

Group- A**Attempt all the questions****[10 x1=10]**

1. Circle (O) the correct answer in the following questions:
 - i. Which of these clauses will be executed even if no exceptions are found?
a) throws b) finally c) throw d) catch
 - ii. Which of these exceptions will occur if we try to access the index of an array beyond its length?
a) ArithmeticException b) ArrayException
c) ArrayArgueumentException d) IndexOutOfRangeException
 - iii. To implement delegates, the necessary condition is?
a) class declaration b) inheritance
c) runtime polymorphism d) exceptions
 - iv. To geerate a simple notification for an object in runtime, the programming construct to be used for implementing this idea?
a) namespace b) interface
c) delegate d) attribute
 - v. Choose the incorrect statement among the following about the delegate?
a) delegates are of reference types b) delegates are object oriented
c) delegates are type safe d) none of the mentioned
 - vi. Which of the following is an incorrect statement about delegate?
a) a single delegate can invoke more than one method
b) delegates could be shared
c) delegates are type safe wrappers for function pointers
d) delegate is a value type
 - vii. Which of these keywords must be used to monitor exceptions?
a) try b) finally c) throw d) catch
 - viii. Which of these keywords is used to manually throw an exception?
a) try b) finally c) throw d) catch

- xi. Assume 2 columns named as Product and Category how can be both sorted out based on first by category and then by product name?
 - a) varsortedProds = _db.Products.OrderBy(c =>c.Category)
 - b) varsortedProds = _db.Products.OrderBy(c =>c.Category) + ThenBy(n =>n.Name)
 - c) varsortedProds = _db.Products.OrderBy(c =>c.Category) . ThenBy(n =>n.Name)
 - d) all of the mentioned
- x. Choose the wrong statement about the LINQ?
 - a) The main concept behind the linq is query
 - b) linq makes use of foreach loop to execute the query
 - c) It is not required that linq should make use of IEnumerable interface
 - d) None of the mentioned

Group B

Attempt any SIX questions.

[6x5=30]

1. Explain DataReader, DataAdapter, Dataset and Datatable with example.
2. Write a program to show insert and delete operation in database.
3. Write a C# program to demonstrate join, concatand union operationin LINQ.
4. What are control statements? Explain different types of control statements with examples in detail.
5. Explain the benefits of using C# language.
6. What is CLR? Explain its features.
7. What is an array? Differentiate rectangular and jagged array with example.

Group C

Attempt any TWO questions.

[2x10=20]

1. (a) Write a C# program to find sum of 20 numbers in an array.
 (b) Create a simple class named Person that contains basic information like name, age, gender, etc. Your class should also contain functions/methods for storing and displaying data.
2. (a) Define event. How can we declare and invoke event? Explain with the help of suitable program.
 (b) What do you mean by anonymous method? Explain with example.
3. What do you mean by static constructor? How it is different than other types of constructors? Explain with example.

MODEL SET 8

Candidates are required to give their answer in their own words as far as practicable.

Group-A

Attempt all the questions

[10 x1=10]

1. Circle (O) the correct answer in the following questions:
 - i. Which among the following is NOT an exception?
 - a) Stack Overflow
 - b) Arithmetic Overflow or underflow
 - c) Incorrect Arithmetic Expression
 - d) All of the mentioned

Group B

Attempt any SIX questions.

[6x5=30]

1. Differentiate ADO and ADO.NET with example.
 2. Explain the process of creating connection in ADO.NET.
 3. Write a C# program to demonstrate where condition in LINQ.
 4. What do you mean by operators? Explain different operators used in C# with examples.
 5. Differentiate parameters pass by value and by reference with example.
 6. Explain operator precedence with example.
 7. What do you mean by interface? Explain use of interface in C# with example.

Group C

Attempt any TWO questions.

$$[2 \times 10 = 20]$$

1. Write a program using ADO.NET to extract name of those students who lives in Kathmandu district, assuming that the student table has four attributes (ID, name, district, and age).

2. (a) What is multiple inheritance? How it is achieved in C#? Explain with suitable example.

(b) What do you mean by interface? Explain use of interface in C# with example.

3. (a) Write a C# program to sort name of 10 persons in alphabetical order.

(b) Write a C# program to add any two matrices.

