Integrated MCA II Semester 4

Subject Name: .NET Programming with C# – Practical **Subject Code:** IMCA13404

Type of course: Professional Core Course

Prerequisite (if any): -

- Concepts of Object Oriented Programming Approach

List of Courses where this course will be prerequisite:

- .NET Technologies

Rationale: .NET Programming will help students to understand the basic concepts of .Net framework and importance of various coding techniques. This course also helps students understand the role of CLR. The students will be able to follow particular programming methodology with .NET Framework for application development

Teaching and Examination Scheme:

TEACHING SCHEME				Theory Marks		Practical Marks		Total
L	T	P	C	TEE	CAT	TEP	CAP	
0	0	4	2	-	-	30	20	50

CAT: Continuous Assessment Theory comprised of CA1 and CA2 **CA1:** Continuous Assessment (assignments/projects/open book tests/closed book tests **CA2:** Sincerity in attending classes/class tests/ timely submissions of assignments/self-learning attitude/solving advanced problems **TEE:** Term End Examination **TEP:** Term End Practical Exam (Performance and viva on practical skills learned in course) **CAP:** Regular submission of Lab work/Quality of work submitted/Active participation in lab sessions/viva on practical skills learned in course

List of Practical:

Sr. No.	Problem Statement
1	Write a console application to display "Hello, World!" using namespaces and the Main method.
2	Create a program that accepts command-line arguments and displays them in the console.
3	Develop a simple interactive program to take user input (e.g., name and age) and display a formatted output.
4	Write a console application to perform arithmetic operations (ADD, SUBTRACT, MULTIPLY, DIVISION, MOD).
5	Create a program to declare a one-dimensional array, sort it, and display its elements.
6	Write a program to manipulate strings (e.g., create, insert, compare, and concatenate strings).
7	Create a class Salary with Basic, TA, DA, and HRA as members. Use a constructor with default values for DA and HRA to calculate the salary of an employee.
8	Demonstrate inheritance with a base class Shape and derived classes Circle, Square, and Rectangle to calculate their areas.
9	Write a program demonstrating polymorphism with method overriding.
10	Write a program to handle exceptions (e.g., DivideByZeroException and IndexOutOfRangeException) with try-catch and finally blocks.
11	Create a Windows Forms application to accept two numbers in Textbox, perform basic arithmetic operations, and display results in a Label.
12	Create a Windows Form that lets the user select their preferences for a subscription. The form includes: 1. A GroupBox containing RadioButtons to choose the subscription type (e.g., Monthly, Yearly). 2. Several CheckBoxes to add optional services (e.g., Email Alerts, SMS Alerts, and Premium Support). 3. A button that, when clicked, displays the user's choices in a message box.
13	Create a Windows Form application that lets a user: 1. Select an item from a ListBox (e.g., a list of destinations). 2. Choose a mode of transport from a ComboBox (e.g., Bus, Train, and Flight). 3. Pick a departure date using a DateTimePicker. 4. Submit the choices and display the selected values in a proper tabular format on new windows form.
14	Develop a Windows Form application to handle mouse and keyboard events (e.g., clicking a button or entering text). 1. A Button that changes its text when clicked using the mouse. 2. A TextBox where users can type, with functionality to display the entered text in a Label upon pressing the Enter key.

	3. Event handlers that detect mouse hover and key press events, showing corresponding messages. 4. A clear layout to demonstrate how user interactions are handled dynamically in a Windows Form.
15	Create a simple menu-driven application with a MenuStrip and ToolStrip in Windows Forms.
16	Write a console application using ADO.NET to connect to a database and display table data. 1. Establishing a connection to a database using a connection string. 2. Executing a SQL query to retrieve data from a specific table (e.g., SELECT * FROM Employees). 3. Reading and displaying the retrieved data in a formatted way using a SqlDataReader. 4. Proper error handling and resource management, including closing the database connection after the operation.
17	Create a program to insert, update, delete, and retrieve data from a database using SqlConnection and SqlCommand. 1. For Insert, the program prompts the user to input values for table columns (e.g., Name, Age, and Email). 2. For Retrieve, it displays all rows in a tabular format with headers. 3. For Update and Delete, it prompts for an identifier (e.g., ID) to modify or remove a specific record, showing a success message for each operation.
18	Develop a program to demonstrate disconnected data access using a DataSet object. 1. A DataGridView control to display data retrieved from a database table (e.g., Products or Employees). 2. A Fetch Data button to populate the DataSet with data from the database using a SqlDataAdapter and display it in the DataGridView. 3. A Save Changes button that applies updates made in the DataGridView back to the database using SqlDataAdapter.Update(). 4. Proper error handling to manage connection issues and validation to ensure data consistency.
19	Create a Windows Forms application to display database records in a DataGridView and allow updates. 1. A DataGridView control to display records from a database table (e.g., Employee details such as ID, Name, Age, and Salary). 2. A Fetch Data button that populates the DataGridView with records from the database. 3. A Save Changes button that updates the database with any edits made directly in the DataGridView. 4. Clear labels and user-friendly layout to ensure ease of interaction and understanding. 5. Also sign an assembly with a strong name.

Suggested Specification table with Marks (Practical):

%Distribution of Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
25	25	50	0	0	0

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy)

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Books:

Sr. no.	Title of book /article	Author(s)	Publisher and details like ISBN	Year of publication	Publication Edition
1	.NET 4.0 Programming (6-in1) (Black Book)	Kogent Learning Solutions Inc	Dreamtech Press ISBN: 978935004 0430	2011	1st edition
2	Pro C# 5.0 and the .NET 4.5 Framework	Andrew Trolsen	WielyAppress	2012	6th edition
3	Programming In C#	E Balagurusamy	McGraw Hill Education	2017	4th edition
4	C# 4.0 The Complete Reference	Herbert Schildt	Osborne/McGraw-Hill	2010	1st edition

Course Outcomes:

Sr. No.	CO Statement After learning this subject, students will be able to	Marks % Weightage
CO-1	Utilize the .NET Framework to create applications that exemplify the architecture and core components of .NET technology.	12
CO-2	Apply the fundamental concepts of C# programming, including namespaces, command-line arguments, and interactive inputs.	18
CO-3	Demonstrate the core programming concepts in C# language, with a deep understanding of Object-Oriented Programming (OOP) principles.	18
CO-4	Develop Windows-based applications using event-driven programming concepts in the .NET environment.	26
CO-5	Implement database connectivity using ADO.NET and effectively manage runtime errors through advanced exception-handling techniques.	26

Mapping with POs:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PO 13
CO-1	3	3	0	1	3	0	0	0	2	0	0	3	0
CO-2	3	3	0	1	3	0	0	0	2	0	0	3	0
CO-3	3	3	0	1	3	0	0	0	2	0	0	3	0
CO-4	3	3	0	1	3	0	3	0	2	0	0	3	0
CO-5	3	3	0	1	3	0	3	0	2	0	0	3	0
Rationale*													

Rationale*: Explaining why it is matching this particular program outcome