

RAN-2311000702020002

M. Sc. (ICT) (Sem.-II) Examination April - 2026 Application Development using .Net Core (Paper-202)

Time: 3 Hours]

[Total Marks: 70

સૂચના : / Instructions (1) ઉપરોક્ત દર્શાવેલ વિગતો ઉત્તરવહી પર અવશ્ય લખવી. Fill up strictly the above details on your answer book. (2) All questions are compulsory. (3) Give examples wherever applicable.	Seat No.: Student's Signature
---	--

- Q. 1. Do as directed. 14**
- Explain the role of Common Language Runtime (CLR) and MSIL in the .NET Core architecture. 05
 - Define Generics. Explain how type constraints (restrictions) improve code safety with an example. 07
 - What is an Assembly? Differentiate between Shared and Private Assemblies. 02
- Q. 2. Answer any TWO from the following. 14**
- Illustrate the use of Multicast Delegates with a C# code snippet.
 - Demonstrate how to implement Interface-based polymorphism to handle different payment methods (e.g., Card, UPI).
 - Create a String Indexer for a "Book" class to find chapters by title.
- Q. 3. Answer any TWO from the following. 14**
- Compare and contrast Class (Reference Type) vs. Structure (Value Type) in terms of memory allocation and performance.
 - Analyze the life cycle of data in ViewData, ViewBag, and TempData. When should each be used?
 - Compare .NET Core and .NET Framework regarding cross-platform support and deployment.
- Q. 4. Write a detail note on any TWO from the following. 14**
- The importance of Dependency Injection (DI) in reducing tight coupling in MVC applications.
 - The use of Action Filters for implementing global logging in a Web application.
 - Entity Framework Core simplifies database integration compared to traditional ADO.NET.
- Q. 5. Answer any TWO from the following. 14**
- Design a custom Event in C# that triggers when a bank balance falls below a specific threshold.
 - Construct a URL Routing configuration in Startup.cs (or Program.cs) that handles a custom "Products/Category/ID" pattern.
 - Develop a sample Web API Controller with GET and POST methods and explain how to test it using Swagger.