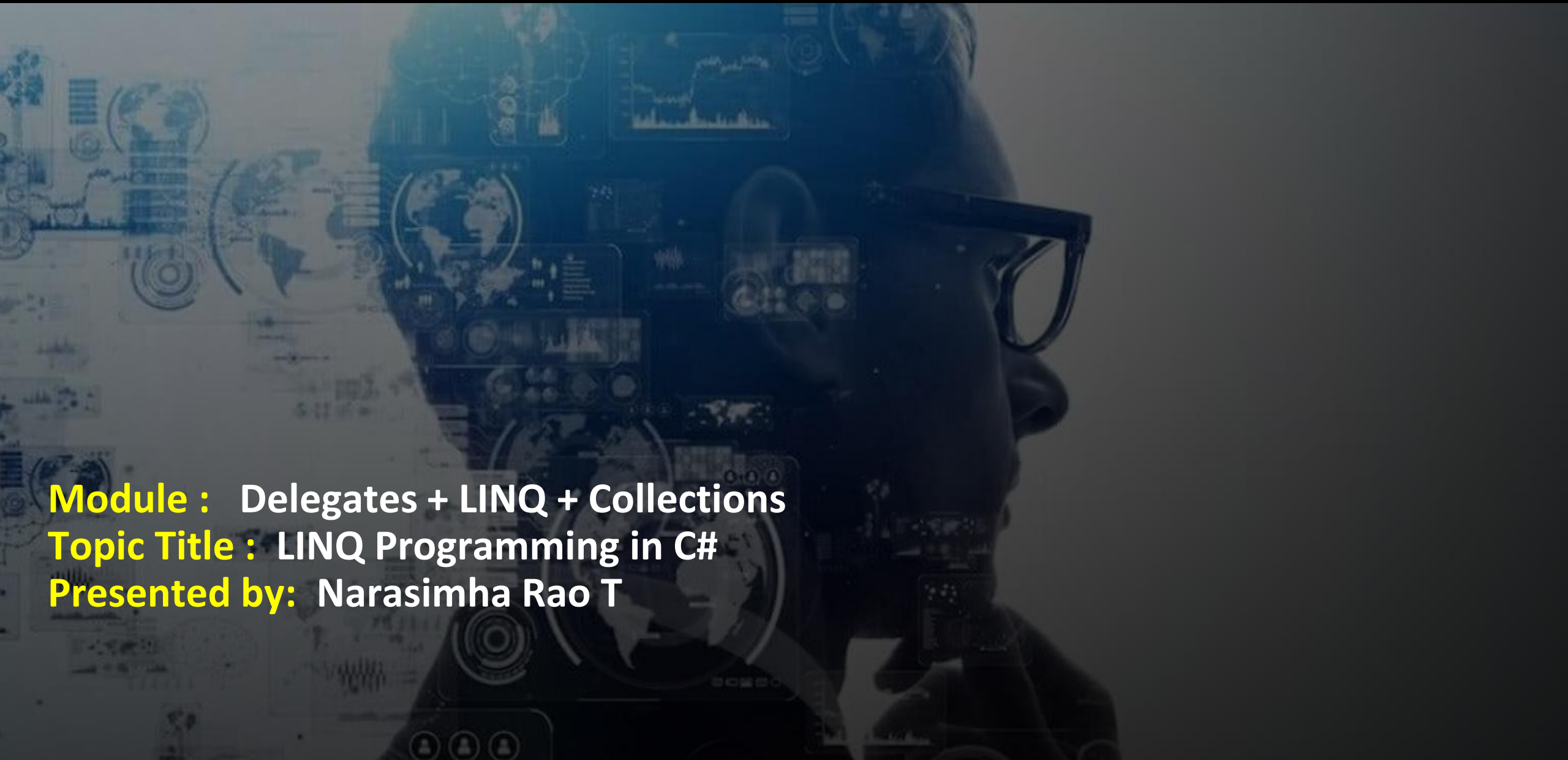

upGrad ENTERPRISE
.NET FSD
Bootcamp Training
23-07-2025

upGrad ENTERPRISE



Module : Delegates + LINQ + Collections
Topic Title : LINQ Programming in C#
Presented by: Narasimha Rao T

upGrad ENTERPRISE

LINQ Programming in C#

By
Narasimha Rao T
Microsoft.Net FSD Trainer
Professional Development Trainer
tnrao.trainer@gmail.com

upGrad ENTERPRISE

Day11

Index

1. Intro to LINQ: Why use it?
2. LINQ Query vs Method syntax
3. Filtering (Where), projection (Select), ordering (OrderBy)
4. Aggregates: Count, Sum, Average, Max, Min
5. Anonymous types
6. Hands-Ons
7. Q & A

Introduciton to LINQ in C#

1. What is LINQ?

LINQ (Language Integrated Query) is a set of features in C# that allows querying data from various sources (collections, databases, XML, etc.) using a unified syntax.

Why Use LINQ?

- **Readable and concise** syntax for querying data.
- **Type safety** at compile-time.
- Reduces **boilerplate code** (like loops).
- Enables querying **in-memory collections** just like database tables.

2. LINQ Query Syntax vs Method Syntax

Query Syntax (SQL-like)

```
var result = from s in students
             where s.Age > 18
             select s.Name;
```

Method Syntax (Lambda expressions)

```
var result = students
             .Where(s => s.Age > 18)
             .Select(s => s.Name);
```

> Both return the same result. Choose based on readability and familiarity.

3. Filtering Data - `Where`

Purpose: Filters a collection based on a condition.

```
var adults = people.Where(p => p.Age >= 18);
```

Example: Filter all students with grade above 70

```
var passedStudents = students.Where(s => s.Grade > 70);
```

4. Projection - `Select`

Purpose: Transform each element of a collection.

```
var names = students.Select(s => s.Name);
```

Example: Project student names and grades

```
var summary = students.Select(s => new { s.Name, s.Grade });
```

5. Ordering - `OrderBy`, `OrderByDescending`

```
var sorted = students.OrderBy(s => s.Grade);
var sortedDesc = students.OrderByDescending(s => s.Grade);
```

Example: Sort products by price

```
var cheapToExpensive = products.OrderBy(p => p.Price);
```

6. Aggregates: Count, Sum, Average, Max, Min

Count

```
int total = students.Count();
```

Sum

```
double totalMarks = students.Sum(s => s.Grade);
```

Average

```
double averageGrade = students.Average(s => s.Grade);
```

Max / Min

```
var topScore = students.Max(s => s.Grade);
var lowestScore = students.Min(s => s.Grade);
```

7. Anonymous Types

Used to create objects without defining a class.

```
var studentSummary = students.Select(s => new
{
    FullName = s.Name,
    Status = s.Grade > 70 ? "Pass" : "Fail"
});
```

> Useful in projection when only part of the object is needed.

Real-Time Examples

Real-Time Examples

E-Commerce: Filter expensive products

```
var premium = products.Where(p => p.Price > 1000);
```

Education App: Top performers

```
var toppers = students
              .OrderByDescending(s => s.Grade)
              .Take(5)
              .Select(s => new { s.Name, s.Grade });
```

Employee Data: Count employees in each department

```
var deptCounts = employees
                 .GroupBy(e => e.Department)
                 .Select(g => new { Department = g.Key, Count = g.Count() });
```

Quiz Time

Quiz Questions

Multiple Choice:

1. What does the `Where` clause do in LINQ?

a) Sorts data

b) Filters data based on condition

c) Aggregates data

d) Projects data

2. Which syntax is SQL-like in LINQ?

a) Query syntax

b) Lambda syntax

c) Method chaining

d) None

3. Which of the following is NOT an aggregate function?

a) Sum

b) Average

c) OrderBy

d) Count

4. What does `Select` do in LINQ?

a) Filters elements

b) Projects each element

c) Removes duplicates

d) Sorts data

Short Answer:

5. Write a LINQ query to get names of employees older than 30.

6. How do you get the total salary paid to employees using LINQ?

upGrad ENTERPRISE

Q & A

Narasimha Rao T
tnrao.trainer@gmail.com