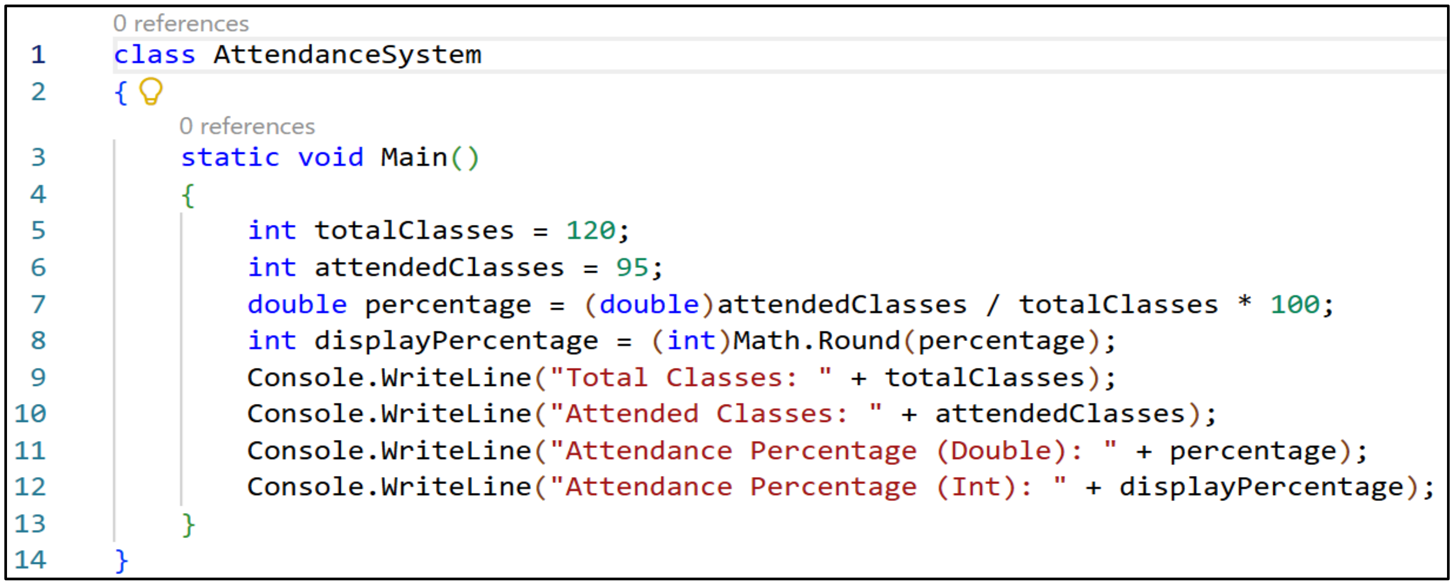
DAY 2 [07-01-2026]

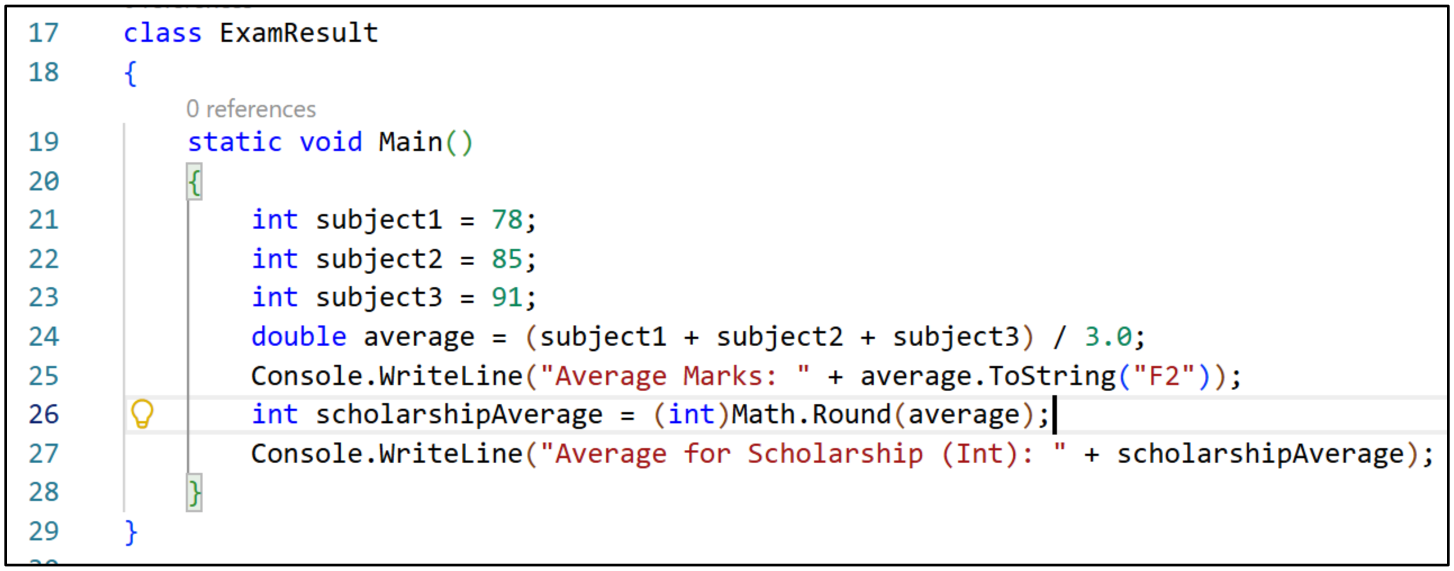
.NET 8/C#12- Detailed Case study Exercise and Value types and Type Conversion:

Exercise 1: Student Attendance and Eligibility System



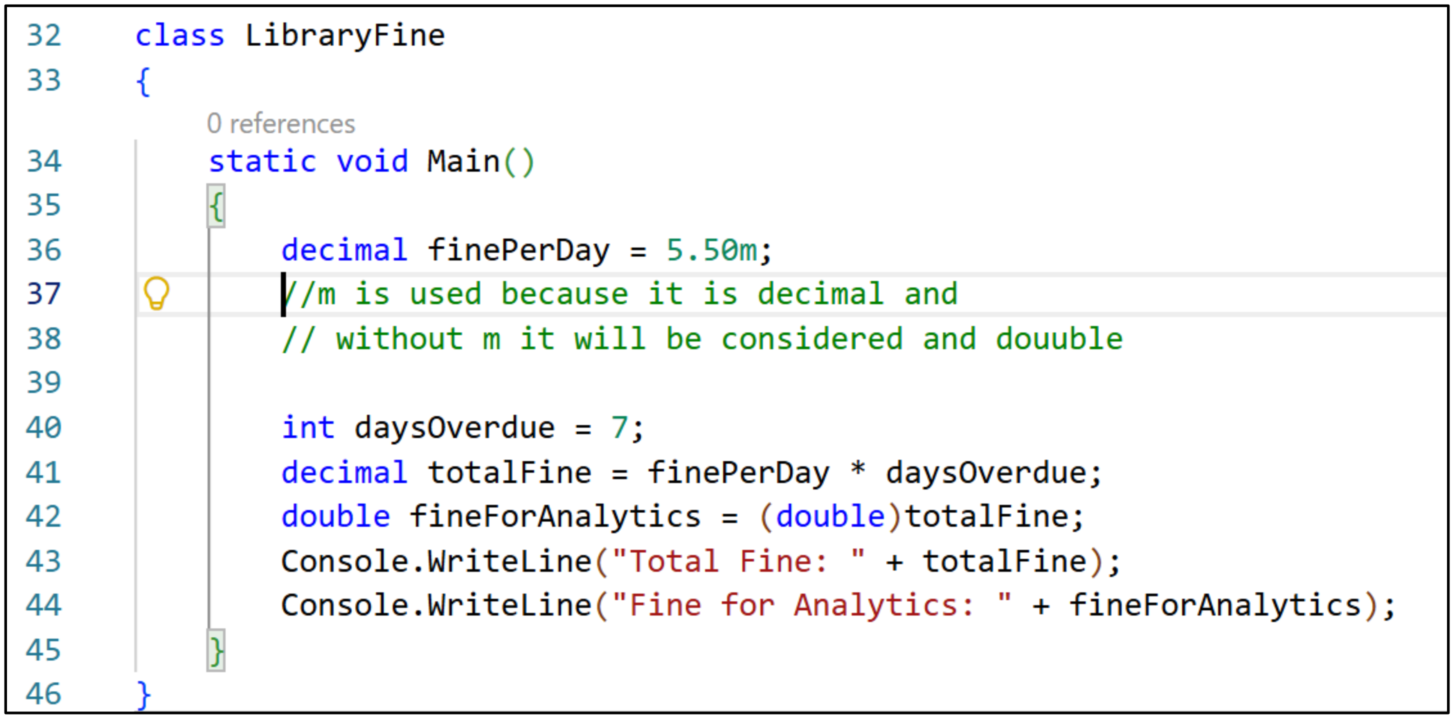
```csharp
0 references
class AttendanceSystem
{
    0 references
    static void Main()
    {
        int totalClasses = 120;
        int attendedClasses = 95;
        double percentage = (double)attendedClasses / totalClasses * 100;
        int displayPercentage = (int)Math.Round(percentage);
        Console.WriteLine("Total Classes: " + totalClasses);
        Console.WriteLine("Attended Classes: " + attendedClasses);
        Console.WriteLine("Attendance Percentage (Double): " + percentage);
        Console.WriteLine("Attendance Percentage (Int): " + displayPercentage);
    }
}
```

Exercise 2: Online Examination Result Processing



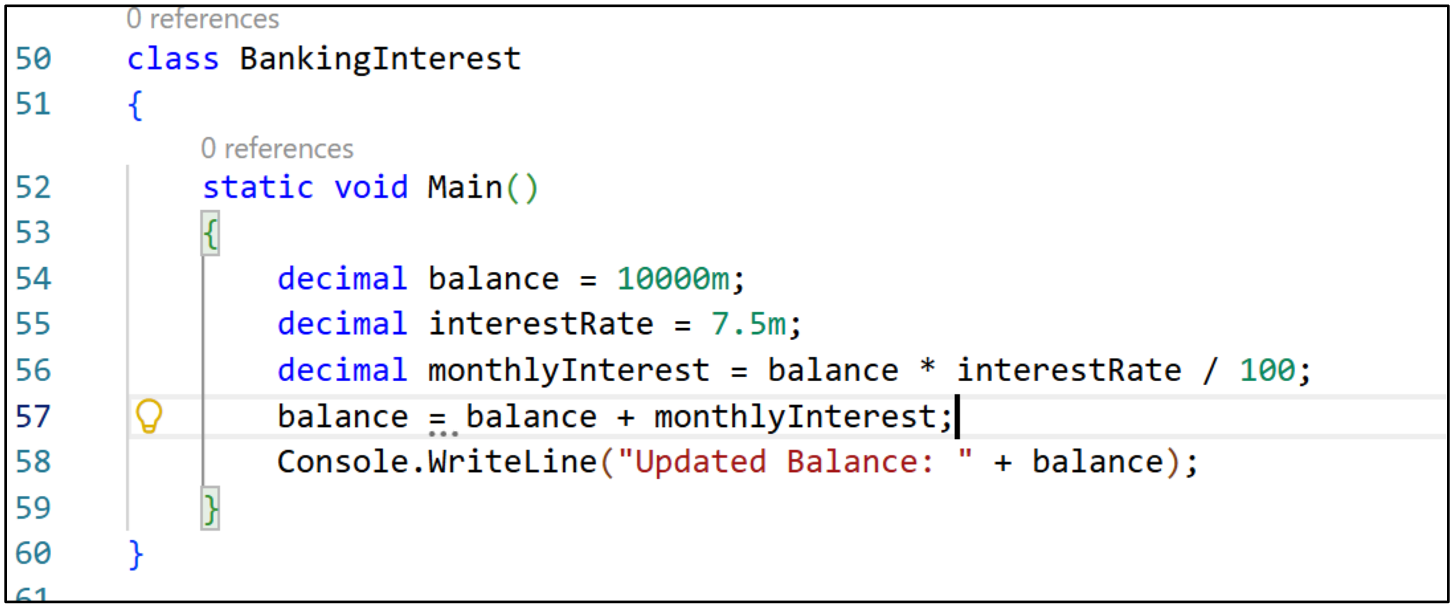
```csharp
class ExamResult
{
    0 references
    static void Main()
    {
        int subject1 = 78;
        int subject2 = 85;
        int subject3 = 91;
        double average = (subject1 + subject2 + subject3) / 3.0;
        Console.WriteLine("Average Marks: " + average.ToString("F2"));
        int scholarshipAverage = (int)Math.Round(average);
        Console.WriteLine("Average for Scholarship (Int): " + scholarshipAverage);
    }
}
```

Exercise 3: Library Fine Calculation Module



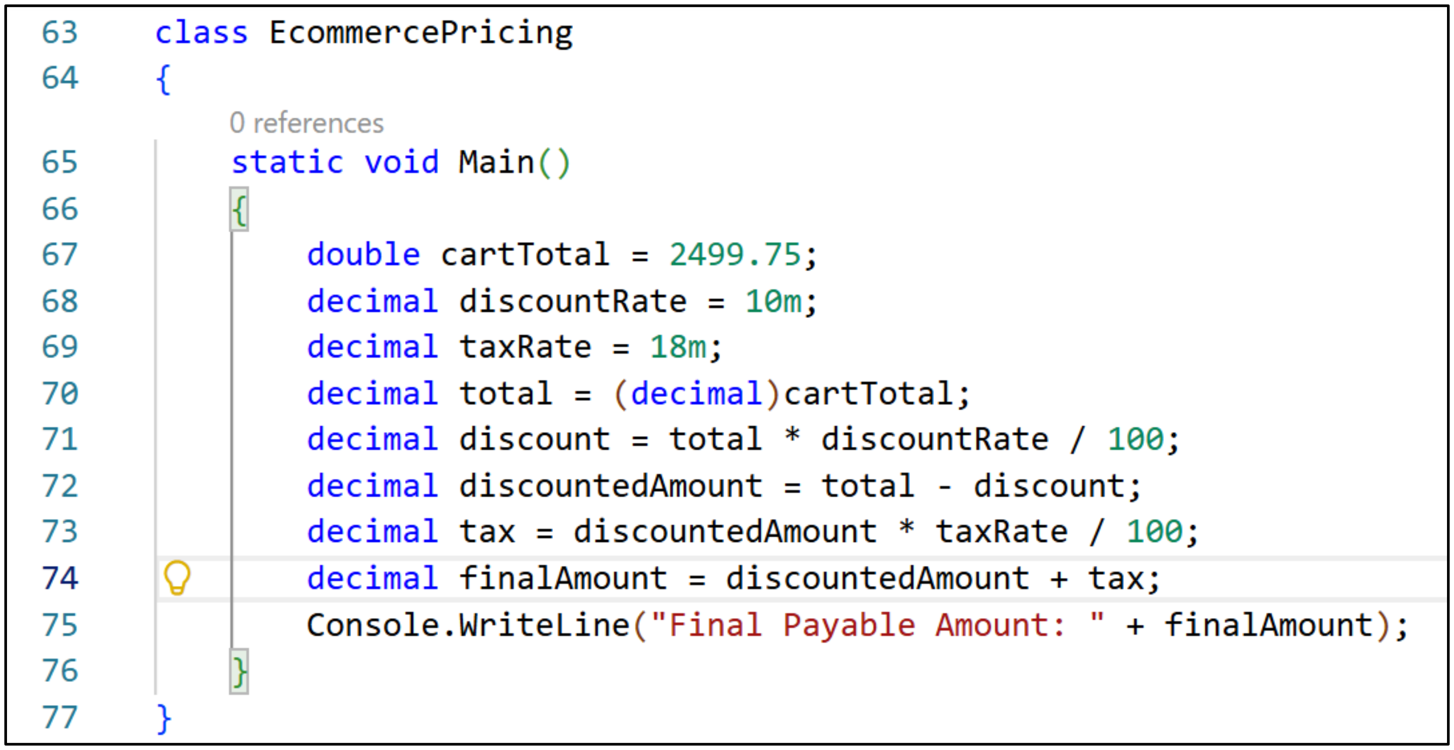
```
class LibraryFine
{
    0 references
    static void Main()
    {
        decimal finePerDay = 5.50m;
        //m is used because it is decimal and
        // without m it will be considered and douuble

        int daysOverdue = 7;
        decimal totalFine = finePerDay * daysOverdue;
        double fineForAnalytics = (double)totalFine;
        Console.WriteLine("Total Fine: " + totalFine);
        Console.WriteLine("Fine for Analytics: " + fineForAnalytics);
    }
}
```

Exercise 4: Banking Interest Calculation Module



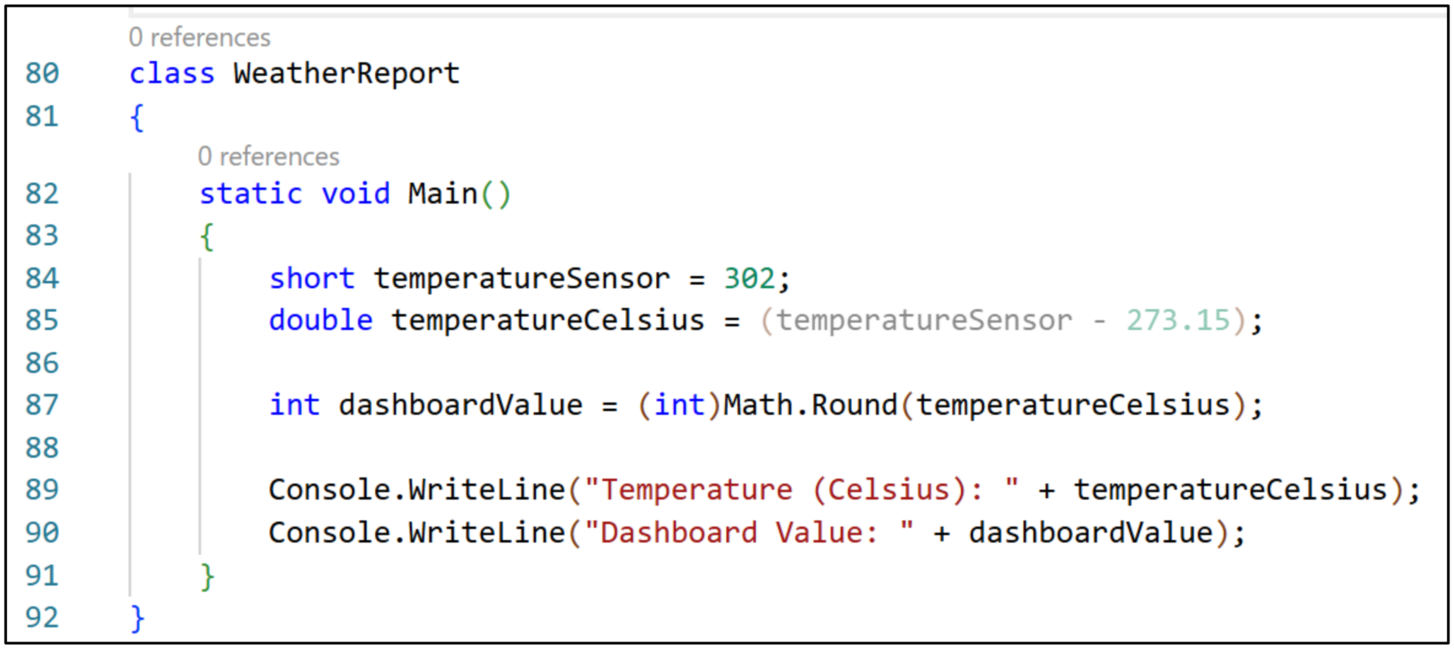
```
0 references
class BankingInterest
{
    0 references
    static void Main()
    {
        decimal balance = 10000m;
        decimal interestRate = 7.5m;
        decimal monthlyInterest = balance * interestRate / 100;
        balance = balance + monthlyInterest;
        Console.WriteLine("Updated Balance: " + balance);
    }
}
```

Exercise 5: E-commerce Order Pricing Engine



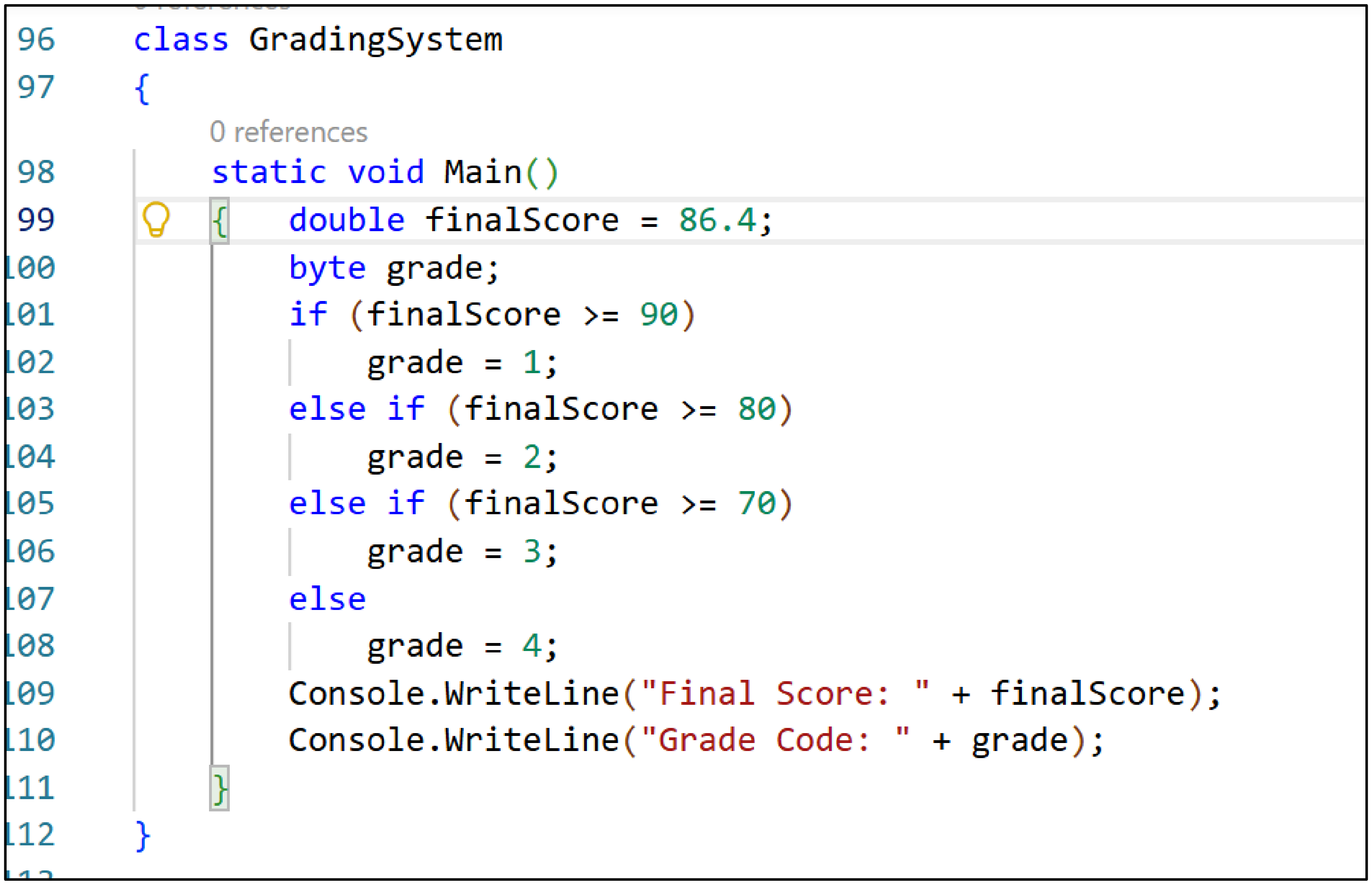
```
class EcommercePricing
{
    0 references
    static void Main()
    {
        double cartTotal = 2499.75;
        decimal discountRate = 10m;
        decimal taxRate = 18m;
        decimal total = (decimal)cartTotal;
        decimal discount = total * discountRate / 100;
        decimal discountedAmount = total - discount;
        decimal tax = discountedAmount * taxRate / 100;
        decimal finalAmount = discountedAmount + tax;
        Console.WriteLine("Final Payable Amount: " + finalAmount);
    }
}
```

Exercise 6: Weather Monitoring and Reporting



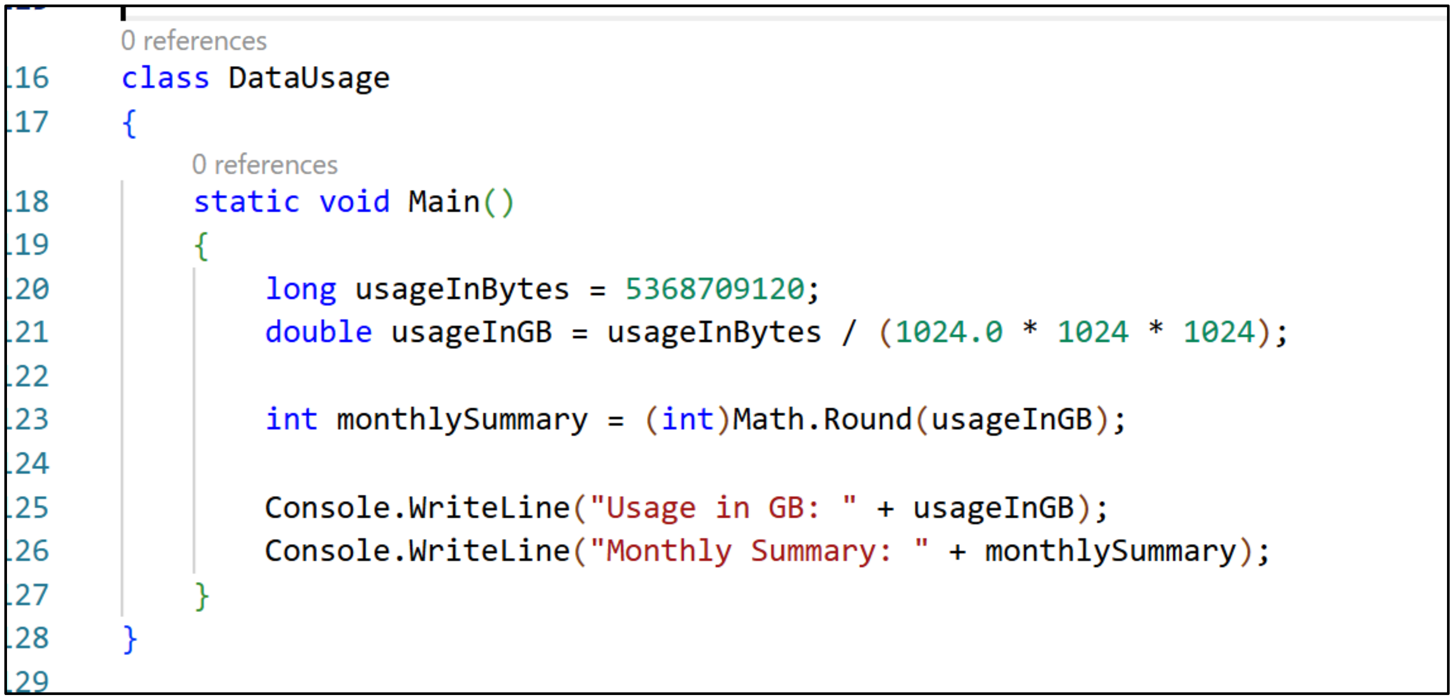
```
0 references
class WeatherReport
{
    0 references
    static void Main()
    {
        short temperatureSensor = 302;
        double temperatureCelsius = (temperatureSensor - 273.15);

        int dashboardValue = (int)Math.Round(temperatureCelsius);

        Console.WriteLine("Temperature (Celsius): " + temperatureCelsius);
        Console.WriteLine("Dashboard Value: " + dashboardValue);
    }
}
```

Exercise 7: University Grading Engine



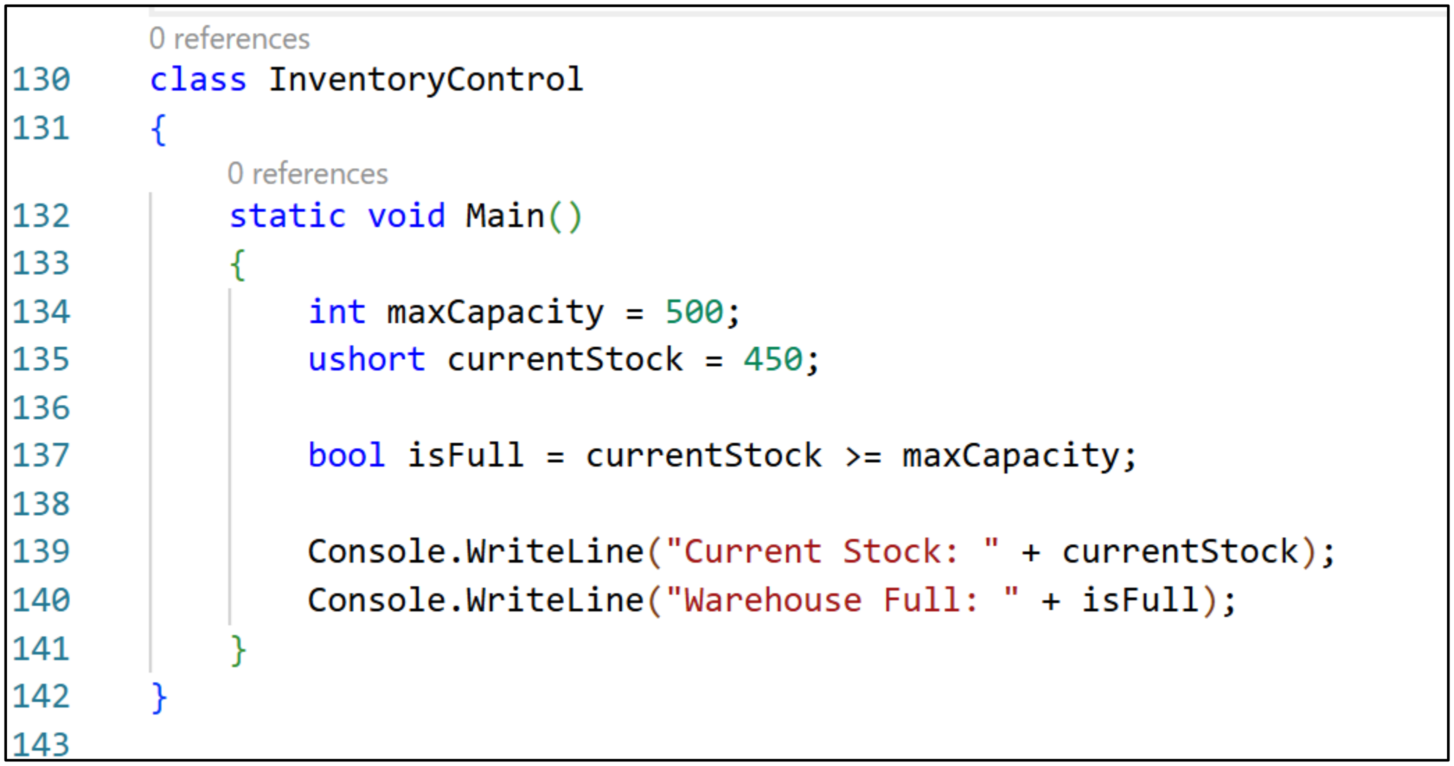
```
class GradingSystem
{
    0 references
    static void Main()
    {   double finalScore = 86.4;
        byte grade;
        if (finalScore >= 90)
            grade = 1;
        else if (finalScore >= 80)
            grade = 2;
        else if (finalScore >= 70)
            grade = 3;
        else
            grade = 4;
        Console.WriteLine("Final Score: " + finalScore);
        Console.WriteLine("Grade Code: " + grade);
    }
}
```

Exercise 8: Mobile Data Usage Trackers



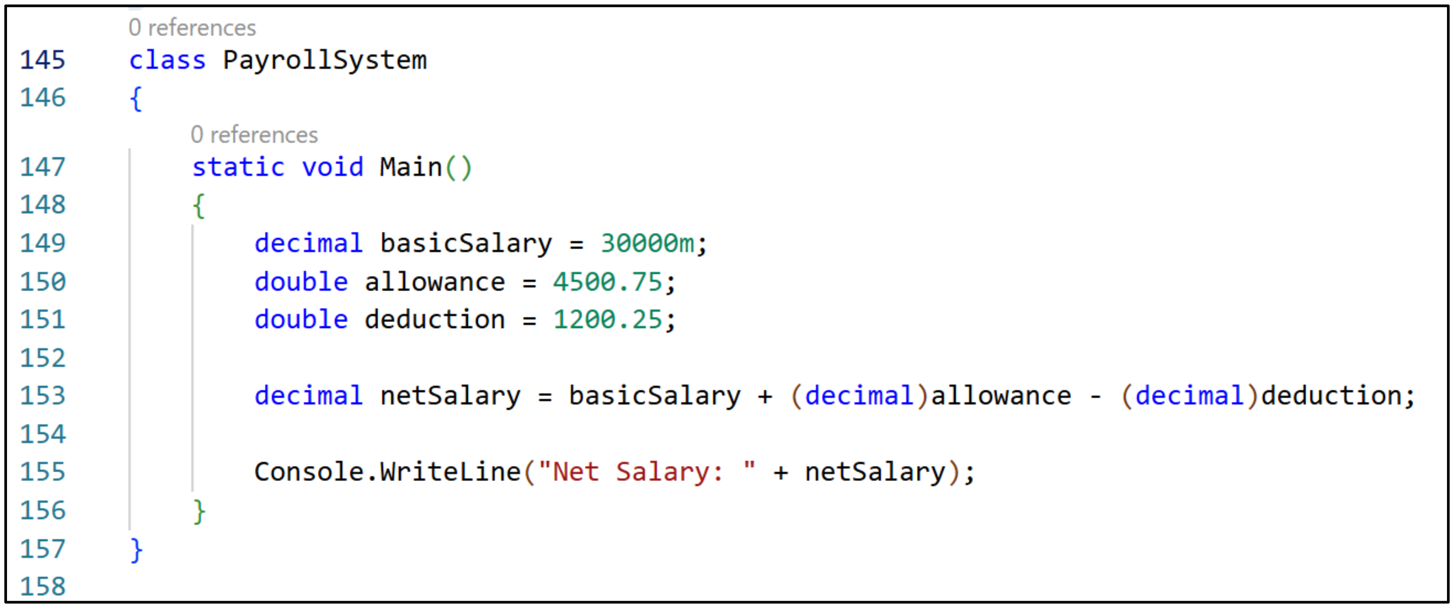
```
0 references
class DataUsage
{
    0 references
    static void Main()
    {
        long usageInBytes = 5368709120;
        double usageInGB = usageInBytes / (1024.0 * 1024 * 1024);

        int monthlySummary = (int)Math.Round(usageInGB);

        Console.WriteLine("Usage in GB: " + usageInGB);
        Console.WriteLine("Monthly Summary: " + monthlySummary);
    }
}
```

Exercise 9: Warehouse Inventory Capacity Control



```
class InventoryControl
{
    static void Main()
    {
        int maxCapacity = 500;
        ushort currentStock = 450;

        bool isFull = currentStock >= maxCapacity;

        Console.WriteLine("Current Stock: " + currentStock);
        Console.WriteLine("Warehouse Full: " + isFull);
    }
}
```

Exercise 10: Payroll Salary Computation



```
class PayrollSystem
{
    static void Main()
    {
        decimal basicSalary = 30000m;
        double allowance = 4500.75;
        double deduction = 1200.25;

        decimal netSalary = basicSalary + (decimal)allowance - (decimal)deduction;

        Console.WriteLine("Net Salary: " + netSalary);
    }
}
```