NAME: AKASH PANDEY

SECTION: 4

ID: 22BCS11135

ASSESSMENT-01

Ans 1:

Method used: Type Casting + Rounding

Explanation: Student Attendance & Eligibility

- Attendance is stored as int because it represents whole numbers.
- Percentage is calculated using double to preserve decimal accuracy.
- Rounding is used before converting to int to avoid wrong eligibility results.

Code:

```
namespace ConsoleApp2
{
   internal class Program
   {
      static void Main(string[] args)
      {
         int netClasses = 180;
         int attendedClasses = 152;
         double percentage = (attendedClasses * 100.0) / netClasses;
         // For display
         int rounded = (int)Math.Round(percentage);   // recommended
         int truncated = (int)percentage;            // not recommended
      }
   }
}
```

Ans 2:

Method used: Explicit Casting

Explanation: Online Examination Result Processing

- Marks are stored as int since exam scores are whole values.
- Average is calculated in double to maintain precision.
- Rounding is applied before converting to int to reduce precision loss.

Code:

```
namespace ConsoleApp2
{
```

```
    internal class Program
    {
            int[] marks = { 78, 82, 91 };
double avg = marks.Average();
double roundedDecimals = Math.Round(avg, 2);
int scholarshipScore = (int)Math.Round(avg);
    }
}
```

Ans 3:

Method used: Explicit Casting

Explanation 3: Library Fine Calculation System

- decimal is used for fine calculation to ensure accurate money values.
- Days overdue are stored as int because they are count-based.
- Fine is converted to double only for analytics purposes.

Code:

```
namespace ConsoleApp2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            decimal finePerDay = 2.50m;
            int daysLate = 6;
            decimal totalFine = finePerDay * daysLate;
            double analyticsValue = (double)totalFine;
        }
    }
}
```

Ans 4:

Method used: Explicit Casting

Explanation 4: Banking Interest Calculation Module

- Account balance is stored as decimal for financial accuracy.
- Interest rate is received as float from an external API.

- Explicit conversion to decimal is required for safe calculations.

Code:

```
namespace ConsoleApp2
{
   internal class Program
   {
      static void Main(string[] args)
      {
         decimal balance = 100000m;
         float rate = 7.5f;
         // explicit conversion required
         decimal interest = balance * (decimal)rate / 100;
         balance += interest;
      }
   }
}
```

Ans 5:

Method used: Type Conversion

Explanation 5: E-Commerce Order Pricing Engine

- Cart total is accumulated using double due to multiple calculations.
- Tax and discount are handled using decimal to avoid rounding errors.
- Final payable amount is stored as decimal for precision.

Code:

```
namespace ConsoleApp2
{
   internal class Program
   {
      static void Main(string[] args)
      {
         double cartTotal = 3499.99;
         decimal tax = 0.18m;
         decimal discount = 0.10m;
         decimal finalAmount =
            (decimal)cartTotal * (1 + tax - discount);
      }
```

```
    }
}
```

Ans 6:

Method used: Type Casting

Exercise 6: Weather Monitoring & Reporting

- Sensor readings are stored as short to save memory.
- Values are converted to double for accurate temperature calculation.
- Result is rounded and converted to int for dashboard display.

Code:

```
namespace ConsoleApp2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            short raw = 325; // example
            double celsius = raw / 10.0;

            int displayTemp = (int)Math.Round(celsius);
        }
    }
}
```

Ans 7:

Method used: Conditional Casting

Explanation: University Grading Engine

- Final score is calculated as double to include fractional marks.
- Grades are stored as byte because of their limited range.
- Conditional checks ensure safe and valid grade assignment.

Code:

```
namespace ConsoleApp2
{
    internal class Program
    {
```

```
        static void Main(string[] args)
        {
            double score = 87.4;
            byte grade;

            if (score >= 90) grade = 10;
            else if (score >= 80) grade = 9;
            else if (score >= 70) grade = 8;
            else grade = 0;
        }
    }
}
```

Ans 8:

Method used: Implicit Conversion + Rounding

Explanation: Mobile Data Usage Tracker

- Data usage is stored as long to support large values.
- Conversion to double allows MB and GB calculations.
- Rounding is used before converting to int for summary display.

Code:

```
namespace ConsoleApp2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            long bytes = 5368709120; // 5 GB
            double gb = bytes / (1024.0 * 1024 * 1024);
            int roundedGB = (int)Math.Round(gb);
        }
    }
}
```

Ans 9:

Metho

d used: Type Promotion (Implicit Comparison)

Explanation: Warehouse Inventory Capacity Control

- Item count uses int while maximum capacity uses ushort.
- Direct comparison avoids unsafe type casting.
- This prevents overflow and signed/unsigned conversion issues.

Code:

```
namespace ConsoleApp2
{
   internal class Program
   {
      static void Main(string[] args)
      {
         int items = 65000;
         ushort maxCapacity = 60000;

         bool exceeded = items > maxCapacity;
      }
   }
}
```

Ans 10

Method used: Explicit Casting

Explanation: Payroll Salary Computation

- Basic salary is stored as int because it is a fixed amount.
- Allowances and deductions use double due to decimal values.
- Net salary is stored as decimal to ensure financial accuracy.

Code:

```
namespace ConsoleApp2
{
   internal class Program
   {
      static void Main(string[] args)
      {
```

```
            int basic = 40000;
            double allowance = 5500.75;
            double deduction = 1200.25;

            decimal netSalary =
                basic + (decimal)allowance - (decimal)deduction;
        }
    }
}
```