12214994SaurabhRana

Title

Price–Quantity Calculation System Using Custom Exceptions

Question No : 1 / 1

Problem Statement

You are required to develop a **C# console application** that simulates a simple **price–quantity calculation system**.
The application allows the user to input the **price** and **quantity** of a product, validates the inputs, and then calculates the **total cost**.

The application should handle various exceptions that may arise during the **input and processing of data**.

Write the solution within the **Program.cs** file.

Requirements

---

1. Class: InvalidPriceException

- Inherits from `SystemException`
- Two constructors:
  - Default constructor with standard message
  - Constructor that accepts a custom message

---

2. Class: InvalidQuantityException

- Inherits from `SystemException`

- Two constructors:
  - Default constructor with standard message
  - Constructor that accepts a custom message

---

3. Class: Program

Main Method Responsibilities

- Reads **price** and **quantity** from console input
- Validates the inputs:
  - Price must be **greater than zero**
  - Quantity must be **greater than zero**
- Throws:
  - `InvalidPriceException` if price is invalid
  - `InvalidQuantityException` if quantity is invalid
- Calculates total cost = `price × quantity`
- Displays total cost **rounded to one decimal place**
- Handles the following exceptions:
  - `InvalidPriceException`
  - `InvalidQuantityException`
  - `FormatException`
  - `Exception` (generic)

Input Format

- First line: **Price** (decimal)
- Second line: **Quantity** (integer)

Output Format

- If price > 0 and quantity > 0:
- `Total cost is {totalCost}`
- If price ≤ 0:
- `Error: Price must be greater than zero.`
- If quantity ≤ 0:
- `Error: Quantity must be greater than zero.`
- If input is not a valid number:
- `Error: Please enter a valid number.`

Test Cases

Test Case ID	Input	Expected Output
TC01	`232.2, 2`	Total cost is 464.4
TC02	`qwerty`	Error: Please enter a valid number.
TC03	`123.3, -1`	Error: Quantity must be greater than zero.
TC04	`-1`	Error: Price must be greater than zero.

Sample Input 1

```
232.2
2
```

Sample Output 1

```
Total cost is 464.4
```

Sample Input 2

```
qwerty
```

Sample Output 2

```
Error: Please enter a valid number.
```

Sample Input 3

```
123.3
-1
```

Sample Output 3

```
Error: Quantity must be greater than zero.
```

Sample Input 4

```
-1
```

Sample Output 4

```
Error: Price must be greater than zero.
```

C# Solution (Program.cs)

```
using System;


class InvalidPriceException : SystemException
{
    public InvalidPriceException()
        : base("Error: Price must be greater than zero.")
    {
    }


    public InvalidPriceException(string message)
        : base(message)
    {
    }
}


class InvalidQuantityException : SystemException
{
    public InvalidQuantityException()
        : base("Error: Quantity must be greater than zero.")
    {
```

```
    }

    public InvalidQuantityException(string message)
        : base(message)
    {
    }
}

class Program
{
    static void Main()
    {
        try
        {
            decimal price = decimal.Parse(Console.ReadLine());

            if (price <= 0)
                throw new InvalidPriceException();

            int quantity = int.Parse(Console.ReadLine());

            if (quantity <= 0)
                throw new InvalidQuantityException();

            decimal totalCost = price * quantity;
```

```
            Console.WriteLine($"Total cost is {Math.Round(totalCost, 1)}");
        }
        catch (InvalidPriceException ex)
        {
            Console.WriteLine(ex.Message);
        }
        catch (InvalidQuantityException ex)
        {
            Console.WriteLine(ex.Message);
        }
        catch (FormatException)
        {
            Console.WriteLine("Error: Please enter a valid number.");
        }
        catch (Exception)
        {
            Console.WriteLine("Error: Please enter a valid number.");
        }
    }
}
```