Connor Perron

BCS426

Prof. Aydin

Spring 2021

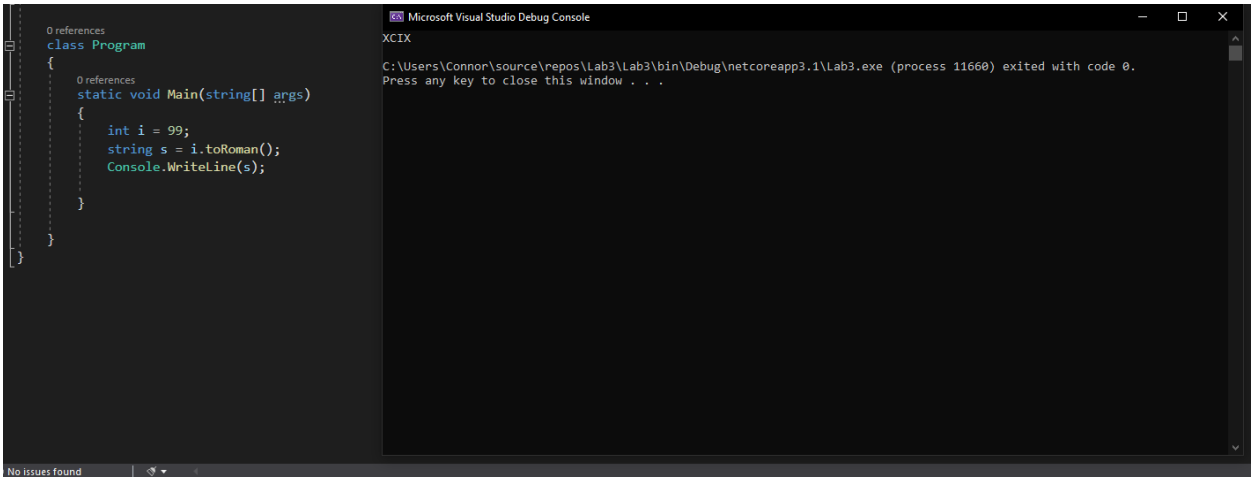
Question 1: Implementing an Extension Method

Your job is to write an extension method called toRoman to convert and int to its roman representation so that a client code such as below would work.

```
static void Main(string[] args)
{
      int i = 99;
      string s = i.toRoman();
       Console.WriteLine(s);

 }
```

```
using Lab3.Roman;
using System;
using System.Collections;

namespace Lab3
{
    namespace Roman
    {
        0 references
        public static class RomanConversion
        {
            1 reference
            public static String toRoman(this int input)
            {
                string numeral = "";

                if (input < 0)
                    return "That is not a valid number";

                else if (input < 1)
                    return "There is no Roman Numberal for 0";

                while (input != 0)
                {

                    if (input >= 1000)
                    {
                       numeral+= "M"; input -= 1000;
                    }

                    else if (input >= 900)
                    {
                        numeral += "CM"; input -= 900;
                    }

                    else if (input >= 900)
                    {
                        numeral += "CM"; input -= 900;
                    }

                    else if (input >= 500)
                    {
                        numeral += "D"; input -= 500;
                    }

                    else if (input >= 400)
                    {
                        numeral += "CD"; input -= 400;
                    }
```

```
        else if (input >= 100)
        {
            numeral += "C"; input -= 100;
        }

        else if (input >= 90)
        {
            numeral += "XC"; input -= 90;
        }

        else if (input >= 50)
        {
            numeral += "L"; input -= 50;
        }

        else if (input >= 40)
        {
            numeral += "XL"; input -= 40;
        }

        else if (input >= 10)
        {
            numeral += "X"; input -= 10;
        }

        else if (input >= 9)
        {
            numeral += "IX"; input -= 9;
        }

        else if (input >= 5)
        {
            numeral += "V"; input -= 5;
        }

        else if (input >= 4)
        {
            numeral += "IV"; input -= 4;
        }

        else if (input >= 1)
        {
            numeral += "I"; input--;
        }
    }
    return numeral;
}
```


0 references
class Program
{
0 references
static void Main(string[] args)
{
int i = 99;
string s = i.toRoman();
Console.WriteLine(s);
}
}
}
Microsoft Visual Studio Debug Console
XCIX
C:\Users\Connor\source\repos\Lab3\Lab3\bin\Debug\netcoreapp3.1\Lab3.exe (process 11660) exited with code 0.
Press any key to close this window . . .
No issues found

QUESTION 2: Abstract classes and Interfaces (Review Week 4 - Chp 4 slides)

- **Create an abstract class Product with following**
  - **Properties Price (Decimal), Code (string), Description (string)**

- **Create an interface ISellable with the following**
  - **Method sell that takes the count of the products to sell and returns the cost**

```
public decimal sell(int count);
```

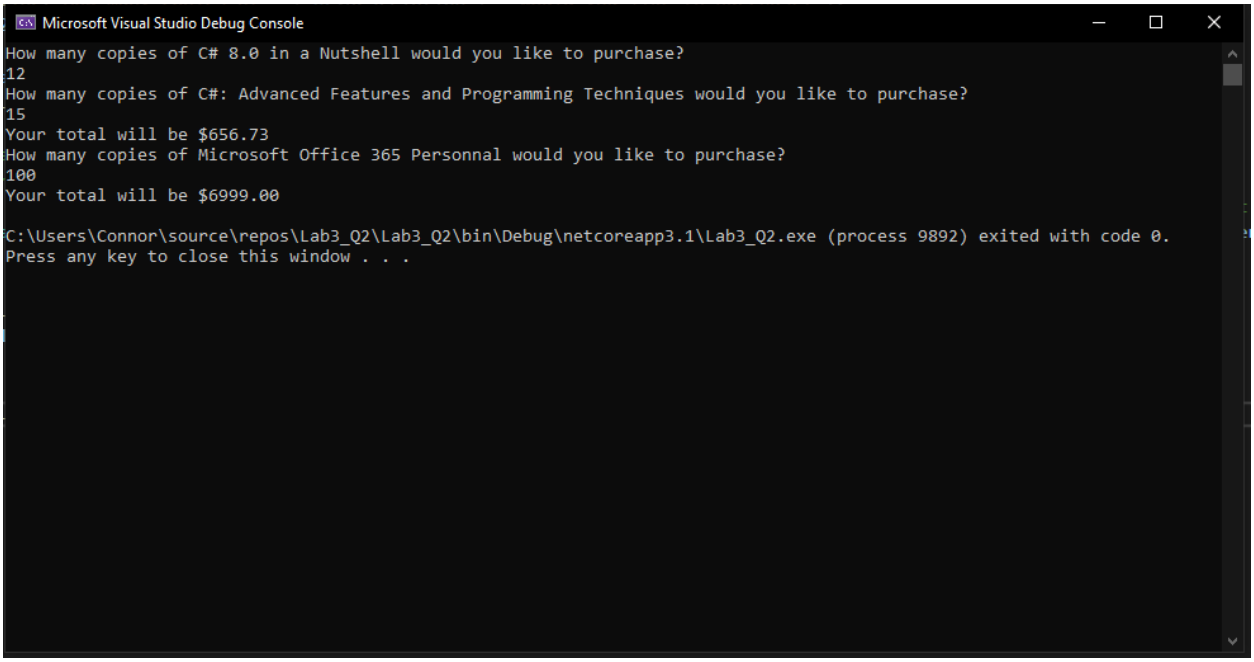
- **Create a concrete class Book that**
  - **Inherits from class Product**
  - **and implements interface ISellable**
  - **Has additional Property Author (string)**
  - **Add a constructor with parameters, and ToString method to display all of data of class Book**
- **Similarly, create anot concrete class Software that**
  - **Inherits from class Product**
  - **and implements interface ISellable**
  - **Has additional Property Version (string)**
  - **Add a constructor with parameters, and ToString method to display all of data of the class Software**
- **Finally, write a client code (main program) that will utilize the class and interface hierarchy you created.**
  - **Create two Book objects for the following books**
    - ***C# 8.0 in a Nutshell, with price $50.99 and with ISBN 978-1492051138 by Jack Smith***
    - ***C#: Advanced Features and Programming Techniques, with price $2.99 and with ISBN 100-1492051000 by Jill Smith***
    - **Ask the user how many of each book the user wants to buy and then display a report of how much the total cost is.**
  - **Create a Software object for the following Software**
    - ***Microsoft 365 Personal with price $69.99 and version 16.0.10827***
    - **Ask the user how many of the Microsoft 365 the user wants to buy and then display a report of how much the total cost is.**

```
public interface ISellable
{
    //  Create an interface ISellable with the following
        //  Method sell that takes the count of the products to sell and returns the cost
    public decimal Sell(int count);
}
```

```
abstract class Product : ISellable
{

    //Create an abstract class Product  with following
        //Properties Price(Decimal), Code(string), Description(string)

    private string _description;
    private decimal _price;
    private string _code;

    public Product (string description, decimal price, string code)
    {
        _description = description;
        _price = price;
        _code = code;
    }

    public string Description { get; set; }
    public string Code { get; set; }
    public decimal Price { get; set; }
    public decimal Sell(int count)
    {
        return count * _price;
    }


}
```

```
class Book : Product, ISellable
{

    private string _author;
    public Book(string description, decimal price, string code, string author) : base(description, price, code)
    {
        this.Description = description;
        this.Price = price;
        this.Code = code;
        _author = author;
    }

    public string Author { get; set; }

    public override string ToString()
    {
        return Description+ " was written by " +_author+ ". It costs " +Price+ " and has a code of "+Code;
    }
}


class Software : Product, ISellable
{
    string _version;
    public Software(string description, decimal price, string code, string version) : base(description, price, code)
    {
        this.Description = description;
        this.Price = price;
        this.Code = code;
        _version = version;
    }

    public string Version { get; set; }

    public override string ToString()
    {
        return Description + " ver. " + _version + ". It costs " + Price + " and has a code of " + Code;
    }
}
```

```
static void Main(string[] args)
{
    //Finally, write a client code(main program) that will utilize the class and interface hierarchy you created.


    //Create two Book objects for the following books
    //C# 8.0 in a Nutshell, with price $50.99 and with ISBN 978-1492051138 by Jack Smith
    //C#: Advanced Features and Programming Techniques, with price $2.99 and with ISBN 100-1492051000 by Jill Smith
    //Ask the user how many of each book the user wants to buy and then display a report of how much the total cost is.
    Book b1 = new Book(description: "C# 8.0 in a Nutshell", price: 50.99m, code: "ISBN 978-1492051138", author: "Jack Smith");
    Book b2 = new Book(description: "C#: Advanced Features and Programming Techniques", price: 2.99m, code: "ISBN 100-1492051000", author: "Jill Smith");

    WriteLine("How many copies of " + b1.Description + " would you like to purchase?");
    int b1Copies = Convert.ToInt32(ReadLine());

    WriteLine("How many copies of " + b2.Description + " would you like to purchase?");
    int b2Copies = Convert.ToInt32(ReadLine());

    decimal total = b1.Sell(b1Copies) + b2.Sell(b2Copies);
    WriteLine("Your total will be $" + total);

    //Create a Software object for the following Software
    //Microsoft 365 Personal with price $69.99 and version 16.0.10827
    //Ask the user how many of the Microsoft 365  the user wants to buy and then display a report of how much the total cost is
    Software s1 = new Software(description: "Microsoft Office 365 Personnal", price: 69.99m, code: "ISBN 100-1492051000", version: "16.0.10827");


    WriteLine("How many copies of " + s1.Description + " would you like to purchase?");
    int s1Copies = Convert.ToInt32(ReadLine());

    total = s1.Sell(s1Copies);
    WriteLine("Your total will be $" + total);



}
```


Microsoft Visual Studio Debug Console
How many copies of C# 8.0 in a Nutshell would you like to purchase?
12
How many copies of C#: Advanced Features and Programming Techniques would you like to purchase?
15
Your total will be $656.73
How many copies of Microsoft Office 365 Personnal would you like to purchase?
100
Your total will be $6999.00
C:\Users\Connor\source\repos\Lab3_Q2\Lab3_Q2\bin\Debug\netcoreapp3.1\Lab3_Q2.exe (process 9892) exited with code 0.
Press any key to close this window . . .