|  |  |  |
| --- | --- | --- |
|  |  | **ISM 6225**  **Distributed Information systems** |

Assignment 1 – Programming Introduction

Primary objective: Develop familiarity with essential programming constructs

Secondary objective: develop comfort with using the IDE

*Estimated time: 10 hours*

## Introduction

Full-stack application development is an essential skill needed to succeed and even survive in business analytics and/ or information systems roles, especially as AI takes over many rudimentary tasks formerly performed by analysts. This assignment introduces the essential programming constructs such as variables, selection, loops, methods and arrays used to build such applications. Specifically, this assignment avoids the use of API methods and object-oriented programming. Those tasks are left for later assignments. This assignment also does not check for efficiency in program implementation. That is something you will develop over a lifetime in the profession. Rather, the focus is on simplicity, and creation of usable programs using industry best practices.

This is an individual assignment, to give every student the opportunity to develop the necessary skills to become a productive contributor to project teams in this class and beyond. One design goal for this assignment was to define an assignment that focused tightly on introductory programming structures, but very low probability that students would find ready-to-use solutions available online.

## Activity

In this assignment, you will define methods to do simple operations. Method signatures and hints are provided below. The methods are listed in the recommended sequence of development. A starter Program.cs file is included in the appendix.

## Submission

Push the code to GitHub and submit the URL. Also, get the output from a sample run that shows the use of all required methods, take the screenshot and upload/push it to GitHub. Finally, add your self-reflection as a comment to the submission.

1. Codes are pushed go GitHub <https://github.com/rkgithubacct/Assignment1_S19.git>
2. Output is uploaded <https://github.com/rkgithubacct/Assignment1_S19/blob/master/OutputScreenShots_Assignment1.JPG>
3. In this exercise, I have learned how to use code repositories in GitHub, git commands using git bash as well as git plug-in with Visual Studio, developed codes using c# and visual studio.

## Grading scheme

Each method carries 4 points. You will be graded on the following aspects:

Logic (including appropriate organization of logic into methods) : 2

Handling all reasonable corner cases : 1

Descriptive comments explaining the logic to reviewer : 1

Self-reflection (learning, recommendations to improve for you, time taken) : 2

## Method specifications

/\*

\* x – starting range, integer (int)

\* y – ending range, integer (int)

\*

\* summary : This method prints all the prime numbers between x and y

\* For example 5, 25 will print all the prime numbers between 5 and 25 i.e.

\* 5, 7, 11, 13, 17, 19, 23

\* Tip: Write a method isPrime() to compute if a number is prime or not.

\*

\* returns : N/A

\* return type : void

\*

\*/

public static void printPrimeNumbers(int x, int y)

/\*

\* para n – number of terms of the series, integer (int)

\*

\* summary : This method computes the series 1/2 – 2!/3 + 3!/4 – 4!/5 --- n \* where ! means factorial, i.e., 4! = 4\*3\*2\*1 = 24. Round off the results to

\* three decimal places.

\* Hint: Odd terms are all positive whereas even terms are all negative.

\* Tip: Write a method to compute factorial of n, call it whenever required.

\*

\* returns : result

\* return type : double

\*/

public static double getSeriesResult(int n)

/\*

\* n – number of lines for the pattern, integer (int)

\*

\* summary : This method prints a triangle using \*

\* For example n = 5 will display the output as:

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*

\* returns : N/A

\* return type : void

\*/

public static void printTriangle(int n)

/\*

\* a – array of elements, integer (int)

\*

\* summary : This method computes the frequency of each element in the array

\* For example a = {1,2,3,2,2,1,3,2} will display the output as:

Number Frequency

1 2

2 4

3 2

\* returns : N/A

\* return type : void

\*/

public static void computeFrequency(int []a)

## Appendix: Program.cs

using System;  
  
namespace Assignment1\_S19  
{  
    class Program  
    {  
        public static void Main()  
        {  
            int a = 5, b = 15;  
            printPrimeNumbers(a, b);  
  
            int n1 = 5;  
            double r1 = getSeriesResult(n1);  
            Console.WriteLine("The sum of the series is: " + r1);

            int n4 = 5;  
            printTriangle(n4);  
  
            int[] arr = new int[] { 1, 2, 3, 2, 2, 1, 3, 2 };  
            computeFrequency(arr);

// write your self-reflection here as a comment

        }  
  
        public static void printPrimeNumbers(int x, int y)  
        {  
            try  
            {  
                // Write your code here  
            }  
            catch  
            {  
                Console.WriteLine("Exception occured while computing printPrimeNumbers()");  
            }  
        }  
  
        public static double getSeriesResult(int n)  
        {  
            try  
            {  
                // Write your code here  
            }  
            catch  
            {  
                Console.WriteLine("Exception occured while computing getSeriesResult()");  
            }  
  
            return 0;  
        }  
  
        public static void printTriangle(int n)  
        {  
            try  
            {  
                // Write your code here  
            }  
            catch  
            {  
                Console.WriteLine("Exception occured while computing printTriangle()");  
            }  
        }  
  
        public static void computeFrequency(int[] a)  
        {  
            try  
            {  
                // Write your code here  
            }  
            catch  
            {  
                Console.WriteLine("Exception occured while computing computeFrequency()");  
            }  
        }  
    }  
}