|  |  |  |
| --- | --- | --- |
|  |  | **ISM 6225**  **Distributed Information systems** |
| Profs Agrawal and Daniel TA: Khaninder Kommagoni | | |

Assignment 1 – Programming Introduction

Primary objective: Develop familiarity with essential programming constructs

Secondary objective: develop comfort with using the IDE and GitHub

*Estimated time: 25 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 simple programming exercises for students to learn basic industry best practices. One design goal for this assignment was to focus tightly on introductory programming structures, with low probability that students would find ready-to-use solutions online.

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.

## Activity

Use a programming language of your choice to define methods to do the operations specified in the method signatures and hints 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 to Canvas. Also, get the output from a sample run that shows the use of all required methods and upload/push a screenshot to GitHub. This serves as a quick check. Submit your self-reflection as a comment to the assignment.

## Grading scheme

Each method carries 1 point. You will be graded on the following aspects for each question:

Logic (including appropriate organization of logic into methods) : 0.5

Handling all reasonable corner cases : 0.2

Descriptive comments explaining the logic to reviewer : 0.2

Self-reflection (time taken, learning, and recommendations) : 0.1

## Method specifications

**QUESTION 1:**

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

\* summary : This method prints number pattern of integers using recursion

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

\* 54321

\* 4321

\* 321

\* 21

\* 1

\* returns : N/A

\* return type : void

\*/

private static void PrintPattern(int n)

**QUESTION 2:**

/\*n2 – number of terms of the series, integer (int)

\* This method prints the following series till n terms: 1, 3, 6, 10, 15, 21……

\* For example, if n2 = 6, output will be

\* 1,3,6,10,15,21

\* Returns : N/A

\* Return type: void

\* Hint: Series is 1,1+2=3,1+2+3=6,1+2+3+4=10,1+2+3+4+5=15, 1+2+3+4+5+6=21……

\*/

private static void PrintSeries(int n2)

**QUESTION 3:**

/\* On planet “USF” which is similar to that of Earth follows different clock

\* where instead of hours they have U , instead of minutes they have S , instead

\* of seconds they have F. Similar to earth where each day has 24 hours, each hour

\* has 60 minutes and each minute has 60 seconds , USF planet’s day has 36 U , each

\* U has 60 S and each S has 45 F.

\* Your task is to write a method usfTime which takes 12HR format and return string

\* representing input time in USF time format.

\* Input format: A string s with time in 12 hour clock format (i.e. hh:mm:ssAM or \* hh:mm:ssPM) where 01<= hh<=12, 00<=mm,ss,<=60

\* Output format: a string with converted time in USF clock format (i.e. UU:SS:FF )

\* where 01<= UU<=36, 00<=SS<=59,00<=FF<=45

\*

\* Sample Input : 09:15:35PM

\* Sample Output: 28:20:35

\*

\* returns : String

\* return type : string

\*

\* Hint: One way of doing this is by calculating total number of seconds in Input time

\* and dividing those seconds according to USF time.

\*/

public static string UsfTime(string s)

**QUESTION 4:**

/\*n- total number of integers( 110 )

\* k-number of numbers per line ( 11)

\* USF Numbers : This method prints the numbers 1 to 110, 11 numbers per line.

\* The method shall print 'U' in place of numbers which are multiple of 3,"S" for

\* multiples of 5,"F" for multiples of 7, 'US' in place of numbers which are multiple

\* of 3 and 5,'SF' in place of numbers which are multiple of 5 and 7 and so on.

\* The output shall look like

\* 1 2 U 4 S U F 8 U S 11

\* U 13 F US 16 17 U 19 S UF 22

\* 23 U S 26 U F 29 US 31 32 U....

\*

\* returns : N/A

\* return type : void

\*/

public static void UsfNumbers(int n3, int k)

**QUESTION 5:**

/\*You are given a list of unique words, the task is to find all the pairs of

\* distinct indices (i,j) in the given list such that, the concatenation of two

\* words i.e. words[i]+words[j] is a palindrome.

\* Example:

\* Input: ["abcd","dcba","lls","s","sssll"]

\* Output: [[0,1],[1,0],[3,2],[2,4]]

\* Explanation: The palindromes are ["dcbaabcd","abcddcba","slls","llssssll"]

\* Example:

\* Input: ["bat","tab","cat"]

\* Output: [[0,1],[1,0]]

\* Explanation: The palindromes are ["battab","tabbat"]

\*

\* returns : N/A

\* return type : void

\*/

public static void PalindromePairs(string[] words)

**QUESTION 6:**

/\*You are playing a stone game with one of your friends. There are N number of

\* stones in a bag, each time one of you take turns to take out 1 to 3 stones.

\* The player who takes out the last stone will be the winner. In this case you

\* will be the first player to remove the stone(s)(Player 1).

\*

\* Write a method to determine whether you can win the game given the number of

\* stones in the bag. Print false if you cannot win the game, otherwise print any

\* one set of moves where you are winning the game. Array should contain moves by

\* both the players.

\* Input: 4

\* Output: false

\* Explanation: As there are 4 stones in the bag, you will never win the game.

\* No matter 1,2 or 3 stones you take out, the last stone will always be removed by \* your friend.

\* Input: 5

\* Output: [1,1,3] or [1,2,2] or [1,3,1]

\* Player 1 picks up 1 stone then player 2 picks up 1 or 2 or 3 stones and the

\* remaining stones are picked up by player 1.

\* Explanation: As there are 5 stones in the bag, you take out one stone.

\* As there are 4 stones in the bag and it’s your friend’s turn. He will never win

\* the game because no matter 1,2 or 3 stones he takes out, you will the one to take

\* out the last stone.

\*

\* returns : N/A

\* return type : void

\*/

public static void Stones(int n4)

Appendix: Program.cs

using System;

namespace Assignment1\_Spring2020

{

class Program

{

static void Main(string[] args)

{

int n = 5;

PrintPattern(n);

int n2 = 6;

PrintSeries(n2);

string s = "09:15:35PM";

string t = UsfTime(s);

Console.WriteLine(t);

int n3 = 110;

int k = 11;

UsfNumbers(n3, k);

string[] words = new string[] { "abcd", "dcba", "lls", "s", "sssll" };

PalindromePairs(words);

}

private static void PrintPattern(int n)

{

try

{

//Write your code here .!

}

catch

{

Console.WriteLine("Exception Occured while computing printPattern");

}

}

private static void PrintSeries(int n2)

{

try

{

//Write your code here .!!

}

catch

{

Console.WriteLine("Exception Occured while computing printSeries");

}

}

public static string UsfTime(string s)

{

try

{

//Write your code here .!!

}

catch

{

Console.WriteLine("Exception Occured while computing UsfTime");

}

return null;

}

public static void UsfNumbers(int n3, int k)

{

try

{

// Write your code here

}

catch

{

Console.WriteLine("Exception occured while computing UsfNumbers()");

}

}

public static void PalindromePairs(string[] words)

{

try

{

// Write your code here

}

catch

{

Console.WriteLine("Exception occured while computing PalindromePairs()");

}

}

public static void Stones(int n4)

{

try

{

// Write your code here

}

catch

{

Console.WriteLine("Exception occured while computing Stones()");

}

}

}

}