CSC180 Assignment #02

**1.** Textbook Exercise 2.3 pages 21-22.

1) Create a new program called Time.cs.

2) Following the example in Section 2.6, create variables named hour, minute and second, and assign them values that are roughly the current time. Use a 24-hour clock, so that at 2pm the value of hour is 14.

int hour, minute, second;

hour = 18;

minute = 42;

second = 10;

3) Make the program calculate and print the number of seconds since midnight.

Console.Write("Number of seconds since midnight: ");

Console.WriteLine(hour \* 60 \* 60 + second);

= 64810

4) Make the program calculate and print the number of seconds remaining in the day.

Console.Write("Number of seconds remaining in the day: ");

Console.WriteLine(86400 - (hour \* 60 \* 60 + second));

= 21590

5) Make the program calculate and print the percentage of the day that has passed.

Console.Write("Percentage of the day that has passed: ");

Console.WriteLine((hour \* 60 \* 60 + second) \* 100 / 86400);

= 75

6) Change the values of hour, minute and second to reflect the current time (I assume that some time has elapsed), and check to make sure that the program works correctly with different values.

It worked on all 3 programs.

**2.** Declare variables using the most appropriate data types (**sbyte, byte, short, ushort, int, uint, long** and **ulong)** in order to assign them the following values:

52,130; -115; 4825932; 97; -10000; 20000; 224; 970,700,000; 112; -44; -1,000,000; 1990; 123456789123456789

https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/integral-numeric-types

ushort a = 52130;

sbyte b = -115;

uint c = 4825932;

byte d = 97;

short e = -10000;

ushort f = 20000;

byte g = 224;

uint h = 970700000;

byte i = 112;

sbyte j = -44;

int k = -1000000;

ushort l = 1990;

ulong m = 123456789123456789;

**3.** Write a program that takes as input a four-digit number in format abcd (e.g. 1234) and performs the following actions: -

Calculates the sum of the digits (in our example 1+2+3+4 = 10).

{

Console.Write("Please type a 4 digit number: ");

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

int firstNumber = (n / 1000) % 10;

int secondNumber = (n / 100) % 10;

int thirdNumber = (n / 10) % 10;

int fourthNumber = (n % 10);

Console.Write("The sum of the number = {0}", firstNumber + secondNumber + thirdNumber + fourthNumber);

Console.ReadLine();

}Prints on the console the number in reversed order e.g., 4321).

{

Console.Write("Please type a 4 digit number: ");

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

int firstNumber = (n / 1000) % 10;

int secondNumber = (n / 100) % 10;

int thirdNumber = (n / 10) % 10;

int fourthNumber = (n % 10);

Console.Write("Reversed number = {0}{1}{2}{3}", fourthNumber, thirdNumber, secondNumber, firstNumber);

Console.ReadLine();

}

Puts the last digit in the first position (in our example 4123).

{

Console.Write("Please type a 4 digit number: ");

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

int firstNumber = (n / 1000) % 10;

int secondNumber = (n / 100) % 10;

int thirdNumber = (n / 10) % 10;

int fourthNumber = (n % 10);

Console.Write("Last digit first = {0}{1}{2}{3}", fourthNumber, firstNumber, secondNumber, thirdNumber);

Console.ReadLine();

}

Exchanges the second and the third digits (in our example 1324).

{

Console.Write("Please type a 4 digit number: ");

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

int firstNumber = (n / 1000) % 10;

int secondNumber = (n / 100) % 10;

int thirdNumber = (n / 10) % 10;

int fourthNumber = (n % 10);

Console.Write("Exchanges second and third = {0}{1}{2}{3}", firstNumber, thirdNumber, secondNumber, fourthNumber);

Console.ReadLine();

}

**Hints:** To get the individual digits of a four-digit number (num) you can use the following expressions: num % 10; (num / 10) % 10; (num / 100) % 10; (num / 1000) % 10;