**Topic: User Defined Data Type**

**Collaboration assignment #1:**  Calculate a person’s Age.

**Points: 30 – 2 bonus points if submitted and working correctly by January 27 2020.**

**Due date**: Thursday, January 31, 2020

Age is a derived attribute. Rather than storing a person’s age which changes every year, it is more practical to store a person’s birthdate and calculate the age based on current date and birthdate.

**What to submit:** project code and a screen shot of at least 5 different sample input/output, one of the input/output combination must be for an invalid input value for one of the age parameters (year, month or day)

**Tasks to be completed:**

Create an abstract data type age. One of the methods will calculate the age of a person. The data for the code can be stored as either structures or classes. This code can be written as a console application in C++ or **similar language** (inlcuidng C, C#, or Java) to meet the following requirements.

1. The program should accept the users input and check for valid/invalid input and warn then the user of any invalid:

Sample input:

What is the First Name: Joe

What is the Last Name: Watson

What is the birth year: 2004

What is the birth month: 7

What is the birth day: 4

1. Use system time to identify today’s date
2. The output should include the person’s name, date of birth, today’s date being used and the person’s age:

**Sample output**:

Joe Watson, age 14 was born on 7-4- 2004 and today is 1-30-2019

1. Use a separate file to store the data as structures (or equivalent constructs- e.g., classes would also be a good choice) to store the person’s information.

The code below is not complete: refer to it if needed. Make sure to organize the code into modules (e.g., methods) also when writing the code, you need to take into account the month and the day. Add validation, and comments to the code.

#include <iostream>

#include <ctime>

using namespace std;

struct perosn

{

char name[20];

int day;

int month;

int year;

}person1;

int main()

{

// current date/time based on current system

time\_t now = time(0);

// convert now to string form

char\* dt = ctime(&now);

cout << "The local date and time is: " << dt << endl;

cout << "Enter age year: ";

cin >> person1.year;

// convert now to tm struct for UTC

tm \*gmtm = gmtime(&now);

dt = asctime(gmtm);

cout << "The UTC date and time is:" << dt << endl;

//cout << "minute :"<< gmtm->tm\_min <<endl;

//int currentYear = (int)gmtm->tm\_year;

int currentYear = gmtm->tm\_year +1900;

cout << "year :" << 1900 + gmtm->tm\_year << endl;

cout << "month :" << gmtm->tm\_mon + 1 << endl;

cout << "day :" << gmtm->tm\_mday - 1 << endl;

char c;

cout << "age: " << currentYear - person1.year;

cin >> c;

}

**Scenario**

1. Ask for the person’s name, and birth date including: the year, month, and day of birth
   1. Make sure the entries are valid for example the month must be an integer between 1 and 12
2. Find current year, month and day
3. Find age = current date- person’s birth date
4. Display the result

Except the provided sample above, the rest of the code must be an original code written by you.