EHB354E - HOMEWORK1

Assignment Date : 21/3/2023
Due Date : 4/4/2023 at 23:59

- Student must do the homework without getting any help or collaboration.
- If significant similarities are found between submitted files, it will be considered as plagiarism, and those homework grades will be zero.
- File should be submitted to Ninova only, email submissions are not accepted.
- Your solution should be in one *.cpp file, without programmer-defined *.h header files.
- Program should work without compiler errors and give correct results as expected.

CUBE CLASS

- Write C++ codes for the UML class diagram given. (+ is public)
- Parametered constructor : Initializes all member datas with parameters.
- volume function : Calculates and returns the volume of Cube.
  Formula = length * width * height
- print function : Displays dimension values and the calculated volume.

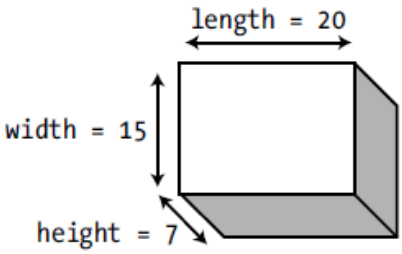
Cube
+ length : int + width : int + height : int
+ Cube () + Cube (int length, int width, int height) + volume() : int + print() : void

NONMEMBER OVERLOADED OPERATOR > FUNCTION

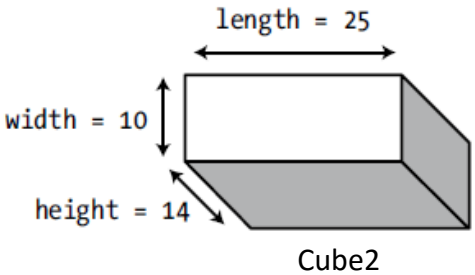
- Write nonmember oveloaded operator > function. Prototype : ***bool operator> (Cube& Cube1 , Cube& Cube2)***
- Function takes two Cubes as arguments and compares calculated volumes of them.
- If volume1 > volume2, then function returns true, otherwise returns false.
- Example calling of operator : if (Cube1 > Cube2) ...

NONMEMBER OVERLOADED OPERATOR + FUNCTION

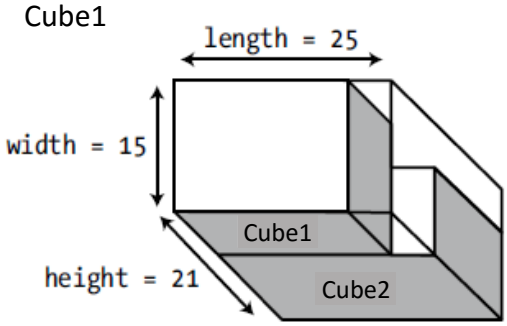
- Write nonmember oveloaded operator + function. Prototype : ***Cube operator+ (Cube& Cube1 , Cube& Cube2)***
- Function takes two Cubes as arguments, adds them and returns a New Cube object.
- The sum of two Cube objects is a New Cube object, that is large enough to contain the two original Cubes stacked on their heights.
- Dimensions of the New Cube is determined as follows.
  New length = Maximum of Length1 and Length2
  New width = Maximum of Width1 and Width2
  New height = Height1 + Height2



Cube1



Cube2



NewCube = Cube1 + Cube2

MAIN PROGRAM

Write a C++ program to do followings. No STL built-in classes or STL functions should be used.

- Define an array of Cube objects (maximum 10 elements), and initialize with constructor parameters below.

Cube No	length	width	height
1	25	56	83
2	5	50	86
3	50	60	76
4	44	35	75
5	40	28	117
6	13	34	95
7	47	32	60
8	43	38	74
9	46	70	78
10	22	26	102

- By calling print function, display all information about the Unsorted Cubes.
- By looping and calling overloaded > operator, sort array of Cubes from smallest volume to biggest volume.
- Display all information about the Sorted Cubes.
- By looping and calling overloaded + operator, build a New Cube object cumulatively, so that all Sorted Cubes are added. Display all information about the New Cube, at each iteration of loop.
- User should not enter any data inputs from keyboard, program assigns all datas through class constructors.
- Use C++ output formatters to obtain column alignments on screen output.

EXAMPLE SCREEN OUTPUT

```
CUBES (UNSORTED)
          L    W    H    Volume
 1.Cube   25   56   83   116200
 2.Cube    5   50   86    21500
 3.Cube   50   60   76   228000
 4.Cube   44   35   75   115500
 5.Cube   40   28  117   131040
 6.Cube   13   34   95    41990
 7.Cube   47   32   60    90240
 8.Cube   43   38   74   120916
 9.Cube   46   70   78   251160
10.Cube   22   26  102    58344

CUBES (SORTED BY VOLUME)
          L    W    H    Volume
 1.Cube    5   50   86    21500
 2.Cube   13   34   95    41990
 3.Cube   22   26  102    58344
 4.Cube   47   32   60    90240
 5.Cube   44   35   75   115500
 6.Cube   25   56   83   116200
 7.Cube   43   38   74   120916
 8.Cube   40   28  117   131040
 9.Cube   50   60   76   228000
10.Cube   46   70   78   251160

CUMULATIVE SUMS OF CUBES AFTER SORTING
                                        L    W    H    Volume
Number of cubes added =  2  , NewCube   13   50  181   117650
Number of cubes added =  3  , NewCube   22   50  283   311300
Number of cubes added =  4  , NewCube   47   50  343   806050
Number of cubes added =  5  , NewCube   47   50  418   982300
Number of cubes added =  6  , NewCube   47   56  501  1318632
Number of cubes added =  7  , NewCube   47   56  575  1513400
Number of cubes added =  8  , NewCube   47   56  692  1821344
Number of cubes added =  9  , NewCube   50   60  768  2304000
Number of cubes added = 10  , NewCube   50   70  846  2961000

PROGRAM FINISHED.
```