#ifndef THROTTLE\_H

#define THROTTLE\_H

/\*  Assignment:     Throttle Class

    Author:         Ryan Wood

    Date Created:   January 13, 2018

    Requirements:   implement the Throttle class from the book

                    with the addition of a maximum throttle amount

\*/

/\*Throttle header file\*/

class Throttle

{

    public:

        /\*  Function:   Throttle()

            Purpose:    default constructor, creates

                        an instance of Throttle with maxPosition

                        set to 6 and curPosition set to 0

            Return:     the Throttle instance

        \*/

        Throttle();

        /\*  Function:   Throttle(int)

            Purpose:    constructor, creates an instance of Throttle with

                        maxPosition set to max ans curPosition set to 0

            Return:     the Throttle instance

        \*/

        Throttle(int);

        /\*  Function:   Throttle(int, int)

            Purpose:    constructor, creates an instance of Throttle with

                        maxPosition set to max and curPosition set to cur

            Parameters: the current position, the max position

            Return:     the Throttle instance

        \*/

        Throttle(int, int);

        /\*  Function:   shutOff()

            Purpose:    sets curPosition to 0

        \*/

        void shutOff();

        /\*  Function:   shift(int)

            Purpose:    increments curPosition by the given amount

            Parameters: the amount to shift

        \*/

        void shift(int);

        /\*  Function:   isOn()

            Purpose:    determines whether the curPosition is

                        greater than zero

            Return:     whether Throttle is on

        \*/

        bool isOn();

        /\*  Function:   flow()

            Purpose:    determines how much throttle there currently

                        is as a percentage of max throttle.

            Return:     the percentage throttle

        \*/

        double flow();

    private:

        int maxPosition;//maximum throttle

        int curPosition;//current throttle

};

#endif

#include "Throttle.h"

#include <iostream>

/\*  Assignment:     Throttle Class

    Author:         Ryan Wood

    Date Created:   January 13, 2018

    Requirements:   implement the Throttle class from the book

                    with the addition of a maximum throttle amount

\*/

/\*Throttle class implementation file\*/

using namespace std;

Throttle::Throttle()

{

    maxPosition = 6;

    curPosition = 0;

}

Throttle::Throttle(int max)

{

    if(max > 0)

        maxPosition = max;

    else

        maxPosition = 1;

    curPosition = 0;

}

Throttle::Throttle(int cur, int max)

{

    if(max <= 0)

        maxPosition = 6;

    else

        maxPosition = max;

    if(cur < 0)

        curPosition = 0;

    else if(cur > max)

        curPosition = max;

    else

        curPosition = cur;

}

void Throttle::shutOff()

{

    curPosition = 0;

}

void Throttle::shift(int amount)

{

    curPosition+= amount;

    if(curPosition > maxPosition)

        curPosition = maxPosition;

    if(curPosition < 0)

        curPosition = 0;

}

bool Throttle::isOn()

{

    return (curPosition > 0);

}

double Throttle::flow()

{

    double dFlow = 0.0;

    dFlow = double (curPosition) / maxPosition;

    dFlow \*= 100;

    return dFlow;

}

/\*  Assignment:     Throttle Class

    Author:         Ryan Wood

    Date Created:   January 13, 2018

    Requirements:   implement the Throttle class from the book

                    with the addition of a maximum throttle amount

\*/

/\*Throttle class test Main file\*/

#include "Throttle.h"

#include <iostream>

#include <iomanip>

#include <fstream>

using namespace std;

ofstream outFile("Throttle.out");

int main()

{

    bool     isOn  = false;

    double   dFlow = 0.0;

    outFile << showpoint << setprecision(4);

    Throttle thCar;

    Throttle thRig(30);

    Throttle thShuttle(4, 20);

    isOn = thCar.isOn();

    outFile << "Car is on before shift = " << isOn << endl;

    thCar.shift(1);

    isOn = thCar.isOn();

    outFile << "Car is on after shift = " << isOn << endl;

    dFlow = thCar.flow();

    outFile << "Car flow at first gear = " << dFlow << "\%" <<endl;

    thCar.shift(4);

    dFlow = thCar.flow();

    outFile << "car flow at 5th gear = " << dFlow << "\%" << endl;

    thCar.shift(-1);

    dFlow = thCar.flow();

    outFile << "Car flow at 4th gear = " << dFlow << "\%" << endl;

    thRig.shift(15);

    dFlow = thRig.flow();

    outFile << "rig flow shifted to 15th gear = " << dFlow << "\%" << endl;

    thRig.shift(20);

    dFlow = thRig.flow();

    outFile << "after shifting up more than rig max flow = " << dFlow << "\%" << endl;

    thRig.shutOff();

    isOn = thRig.isOn();

    outFile << "after shutting rig off, isOn = " << isOn << endl;

    isOn = thShuttle.isOn();

    dFlow = thShuttle.flow();

    outFile << "Shuttle flow at 4th gear is " << dFlow << "\%" << endl;

    outFile.close();

    return 0;

}