//clock1.h, the header file for the class clockType

class clockType
{
private:
    int hr;    //variable to store the hours
    int min;   //variable to store the minutes
    int sec;   //variable to store the seconds

public:
    clockType();
    //Default constructor
    //Sets the clock to 00:00:00
    //Postcondition: hr = 0; min = 0; sec =0.

    clockType(int h, int m, int s);
    //Constructor with parameters
    //The time is set according to the parameters h, m and s.
    //Postcondition: hr = h; min = m; sec =s.

    void setTime(int hours, int minutes, int seconds);
    //Function to set the time.
    //The time is set according to the parameters.
    //Postcondition: hr = hours; min = minutes;
    //sec = seconds
    //The function checks whether the values of
    //hours, minutes, and seconds are valid. If a
    //value is invalid, the default value 0 is
    //assigned.

    int getHour();
    //Function to return the present hour.

    int getMinute();
    //Function to return the present minute.

    int getSecond();
    //Function to return the present second.

    void printTime() const;
    //Function to print the time.
    //Postcondition: The time is printed in the form
    //hh:mm:ss.

    void incrementSeconds();
    //Function to increment the time by one second.
    //Postcondition: The time is incremented by one
    //second.
    //If the before-increment time is 23:59:59, the
    //time is reset to 00:00:00.

    void incrementMinutes();
    //Function to increment the time by one minute.
    //Postcondition: The time is incremented by one
    //minute.
    //If the before-increment time is 23:59:53,
    //the time is reset to 00:00:53.
void incrementHours();
    // Function to increment the time by one hour.
    // Postcondition: The time is incremented by one
    // hour.
    // If the before-increment time is 23:45:53, the
    // time is reset to 00:45:53.

bool equalTime(clockType& otherClock);
    // Function to test whether or not otherClock is
    // giving the same reading as the present clock

};

// Non-member functions
bool isEqual(clockType&, clockType&);
    // Function to test whether or not two clocks are giving
    // the same reading.
    // Similar to the member function equalTime

bool operator == (clockType&, clockType&);
    // Operator overloading to test whether or not two clocks
    // are giving the same reading

ostream& operator << (ostream&, clockType &c);
    // Operator overloading to print the present reading
    // of the clock