

upDate.h

```
/*
 * upDate.h
 *
 * Created on: Sep 18, 2016
 * Author: Kenny Do
 */

#ifndef UPDATE_H_
#define UPDATE_H_

#include <iostream>
#include <string>

class upDate {
    /**
     * Stores the amount of date objects
     */
    static int numOfDateObjects;

    /**
     * A pointer to an array of int
     */
    int * iptr;

    /**
     * Precondition:
     *     julian is positive
     * Postcondition:
     *     Set the date of this upDate object with julian
     */
    void updateFromGregorian(int julian);

    /**
     * Precondition:
     *     None
     * Postcondition:
     *     Sets the date to May 11, 1959
     */
    void defaultDate();

public:
    /**
     * Precondition:
     *     None
     * Postcondition:
     *     Creates a upDate with the default
     */
    upDate();

    /**
     * Precondition:
     *     None
     * Postcondition:
     *     Creates a upDate with M, D, Y
     */
    upDate(int M, int D, int Y);
```

upDate.h

```
/**
 * Precondition:
 *     None
 * Postcondition:
 *     Creates a clone of date
 */
upDate(const upDate& date);

/**
 * Precondition:
 *     None
 * Postcondition:
 *     Destructor for update
 */
~upDate();

/**
 * Precondition:
 *     None
 * Postcondition:
 *     Sets the date to M, D, Y for month, day, year respectively
 */
void setDate(int M, int D, int Y);

/**
 * Precondition:
 *     None
 * Postcondition:
 *     Calculates the days between D and this date
 */
int daysBetween(upDate D);

/**
 * Precondition:
 *     None
 * Postcondition:
 *     Returns the month
 */
int getMonth() const;

/**
 * Precondition:
 *     None
 * Postcondition:
 *     Returns the day
 */
int getDay() const;

/**
 * Precondition:
 *     None
 * Postcondition:
 *     Returns the year
 */
int getYear() const;

/**
```

upDate.h

```
* Precondition:
*     None
* Postcondition:
*     Returns the number of upDate objects
*/
static int GetDateCount();

/**
* Precondition:
*     None
* Postcondition:
*     Returns the month same as a string
*/
std::string getMonthName();

/**
* Precondition:
*     None
* Postcondition:
*     Return the julian number
*/
double julian() const;

/**
* Precondition:
*     julian is positive
* Postcondition:
*     Returns a upDate object that is calculated from the julian number
*/
static upDate returnGregorian(int julian);

/**
* Precondition:
*     None
* Postcondition:
*     Overloads the << operator with the to string of this class
*/
friend std::ostream& operator<<(std::ostream& os, const upDate& date);

/**
* Precondition:
*     None
* Postcondition:
*     Assign this upDate to date
*/
void operator =(const upDate & date);

/**
* Precondition:
*     None
* Postcondition:
*     Adds an int + upDate
*/
friend upDate operator +(int N, const upDate& date);

/**
* Precondition:
```

upDate.h

```
*      None
* Postcondition:
*      Post-increment the date by 1
*/
upDate& operator ++();

/**
 * Precondition:
 *      None
 * Postcondition:
 *      Pre-increment the date by 1
 */
upDate operator ++(int);

/**
 * Precondition:
 *      None
 * Postcondition:
 *      Post-decrement the date by 1
 */
upDate& operator --();

/**
 * Precondition:
 *      None
 * Postcondition:
 *      Pre-decrement the date by 1
 */
upDate operator --(int);

/**
 * Precondition:
 *      None
 * Postcondition:
 *      Increment the date by N
 */
upDate operator +(int N);

/**
 * Precondition:
 *      None
 * Postcondition:
 *      Decrement the date by N
 */
upDate operator -(int N);

/**
 * Precondition:
 *      None
 * Postcondition:
 *      Decrement the date by another date
 */
int operator -(const upDate& date);

/**
 * Precondition:
 *      None
```

upDate.h

```
* Postcondition:
*   Returns true if this date is greater than the other date else false
*/
bool operator >(const upDate & date);

/**
 * Precondition:
 *   None
 * Postcondition:
 *   Returns true if this date is less than the other date else false
 */
bool operator <(const upDate & date);

/**
 * Precondition:
 *   None
 * Postcondition:
 *   Returns true if this date is greater than or equal to than the other date else false
 */
bool operator >=(const upDate & date);

/**
 * Precondition:
 *   None
 * Postcondition:
 *   Returns true if this date is less than or equal to than the other date else false
 */
bool operator <=(const upDate & date);

/**
 * Precondition:
 *   None
 * Postcondition:
 *   Returns true if this date is equal to the other date else false
 */
bool operator ==(const upDate & date);

/**
 * Precondition:
 *   None
 * Postcondition:
 *   Returns true if this date is not equal to the other date else false
 */
bool operator !=(const upDate & date);

};

#endif /* upDate_H_ */
```