Background

You are a new programmer with the Mine Safety and Health Administration (MSHA). Details about mining accidents are recorded locally in text files and then consolidated centrally in one large file. MSHA uses the data in this consolidated file for analysis. There are issues with the data and with existing analysis software. The agency needs to streamline its accident analysis processes.

Your objective is to develop object-oriented software that will load and manipulate the mining accident data. The text files include the following data elements:

```
document_no <unique key>, mind_id, coal_or_metal, controller_id, accident_dt, accident_time,
degree_injury_cd, occupation_cd, no_injuries, return_to_work_dt, trans_term, narrative
```

All data are stored in the file `MineAccidentLog.dat`, a sample of this data is shown below and the complete file may be downloaded from Blackboard (some narrative fields have been shortened to present each record of the sample data on one line).

```
220123250020 3304520 C C02880 2012-11-02 12:00 04 145 1 2013-02-07 N Employee was working on a dryer, the employee stated that he slipped and caught his foot and fell…
220120790037 3601761 C C15935 2012-03-14 08:10 06 168 1 2012-03-15 N While attempting to close the operator's cab door on the H285S Hydraulic Shovel employee pinched his left …
220122650032 3601761 C C15935 2012-09-12 18:45 09 199 1 2012-09-13 N A 15 year old juvenile fell and injured his left ankle while tresspassing on an inactive part of the mine site.
220123120003 3602073 C C09456 2012-10-11 17:15 03 116 1 2012-10-22 N Employee fell down office trailer outside stairs.
220121850008 3605018 C 0041211 2012-06-24 07:00 04 069 1 2013-04-15 N Getting off a 20 ton motor he stepped in a hole and twisted his right knee straining it. He began losing time …
220120650019 3605466 C 0041211 2012-02-21 02:30 03 016 1 2012-03-28 N The employee was attempting to push the edge of a large rock into the feeder after it had been loaded by …
```

Requirements

To begin, you should implement a `Date` class and a `Time` class. There are two date fields and one time field in each record. The first date and the time indicate when the accident occurred. The second date indicates when the employee returned to work after the accident. The `Date` class is especially important and includes several overloaded operators that must be correctly implemented. The `Time` class is important, but we will reuse an existing implementation from a previous MSHA project.

Once your `Date` class is complete, you will need to implement an `MArecord` class. The `MArecord` class should include data members to store each of the data elements in the input data log file. Each instance of the `MArecord` class represents one row from the data file. Overloaded versions of the stream extraction and stream insertion operators must also be included in this class.

Finally, you should implement a class named `MAlog` to store instances of the `MArecord` class. This class shall include a vector of `MArecord` objects. It shall also include member functions to compute various report totals. UML diagrams for these classes are shown below. Implement these classes completely as shown.

Once all classes have been fully implemented and tested; you must demonstrate the functionality of your software. This shall be done by instantiating an object of the `MAlog` class and reading all data from the `MineAccidentLog.dat` file. Invoke the member function `outputRecords` to output the same data read in to a new file with a different name. The output file shall have the same data in the exact same format. Your software shall also display a report on screen similar to the following:

```
number of accident events logged:            19599
number of events resulting in no injuries:   0
number of events resulting in permanent disability: 151
number of events resulting in diagnosis of occupational illness:   239
number of events resulting in death or injury due to natural causes:       25
```

This tests only the basic functionality. During grading numerous other tests will be completed.

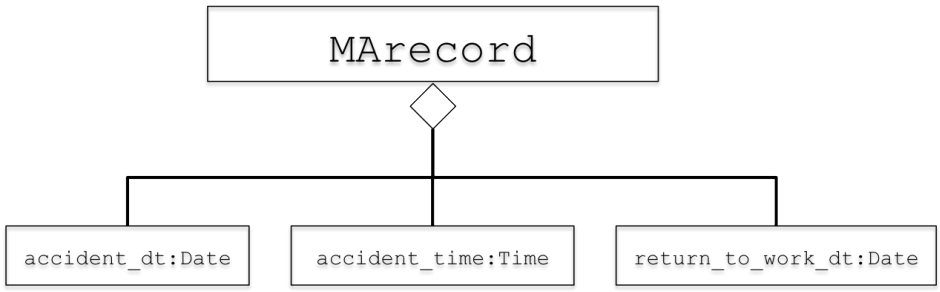
UML Diagrams

Date
- mm : int - dd : int - yyyy : int
+ Date(int = 1923, int = 1, int = 1) //yyyy, mm, dd + Date(const Date&) + setDate(int, int, int) : void //yyyy, mm, dd + setDate(const Date&) : void + getDay() const : int + getMonth() const : int + getYear() const : int + operator <<(ostream&, const Date&) : ostream& + operator >>(istream&, Date&) : istream& + operator ==(const Date&) const : bool + operator !=(const Date&) const : bool + operator <=(const Date&) const : bool + operator <(const Date&) const : bool + operator >=(const Date&) const : bool + operator >(const Date&) const : bool + operator =(const Date&) : const Date&

Time
- hh : int - mm : int
+ Time(int = 0, int = 0) + setTime(int, int) : void + getHour() const : int + getMinute() const : int + getSecond() const : int + operator =(const Time&) : const Time&

MArecord
- document_no : string - mine_id : string - coal_or_metal: char - controller_id : string - accident_dt : Date - accident_time : Time - degree_injury_cd : string - occupation_cd : string - no_injuries : int - return_to_work_dt : Date - trans_term : boolean - narrative : string
+ MArecord(std::string = "", std::string = "", char = '-', std::string = "", Date = Date(1865, 1, 1), Time = Time(0, 0), std::string = "", std::string = "", int = 0, Date = Date(1865, 1, 1), bool = false, std::string = "") + getDocNumber() const : std::string + getMineID() const : std::string + getCoalOrMetal() const : char + getControllerID() const : std::string + getAccidentDate() const : Date + getAccidentTime() const : Time + getInjuryCode() const : std::string + getOccupationCode() const : std::string + getNumberOfInjuries() const : int + getReturnDate() const : Date + getTransOrTerm() const : bool + getNarrative() const : string + operator <<(ostream&, const MArecord&) : ostream& + operator >>(istream&, MArecord&) : istream&

MAlog
- accidents : vector<MArecord> - inputRecords(std::string) : void
+ MAlog(std::string = "") + outputRecords(std::ostream &os = std::cout) : void + getAccidentCount() const : int + getAccidentCount(std::string specificInjuryCode) const : int



Programming Skills

The programming skills required to complete this assignment include:

- Object oriented design
- Class composition
- Function overloading
- Operator overloading
- File input / output

Submission Details

What to submit: One compressed file containing all source code and any other files associated with this project. The file name should be `<netID>P1.zip`.
You must separate your class specification details from your class implementation details. Therefore, you must prepare a header file (`<filename>.h`) and an implementation file (`<filename.cpp>`) for each class. Ensure that your `.h` files contain sufficient comments for each data member and class method. Additionally, you must provide another `.cpp` file that contains function `main()`. This "driver" program is where class objects are instantiated and functionality of the software is demonstrated. Use the following file names:

```
DateTime.h
DateTime.cpp
MArecord.h
MArecord.cpp
MAlog.h
MAlog.cpp
<netID>P1.cpp
```

Due date/time: Thursday, 30 JAN 2013, no later than 11:00 am. Late submissions will be penalized 2.5 points for each 15 minutes late. If you are over 10 hours late you may turn in the project to receive feedback but the grade will be zero. In general requests for extensions will not be considered.

Point Value: 100 points

Academic Integrity

This is an individual project and all work must be your own. Refer to the guidelines specified in the *Academic Honesty* section of this course syllabus or contact me if you have any questions.

Include the following comments at the start of your program:

```
/*
 * <FileName>.<file extension>
 *
 *  COSC 052 Spring 2014
 *  Project #1
 *
 *  Due on: JAN 30, 2014
 *  Author: <your name>
 *
 *
 *  In accordance with the class policies and Georgetown's
 *  Honor Code, I certify that, with the exception of the
 *  class resources and those items noted below, I have neither
 *  given nor received any assistance on this project.
 */
```

Grading

This graded assignment is worth 100 points and will be counted as part of the *Programming Projects* category for the course. Your final score is based on common deductions, as well as, a detailed rubric of points. The table below lists common deductions.

Common Deductions	
Program does not compile	-35.00
Program compiles but has warnings (deduction varies depending on how bad, value listed is max)	-15.00
Program crashes during execution (deduction varies depending on how bad, value listed is max)	-25.00
Filenames do not follow conventions specified (deduction varies depending, value listed is max)	-30.00
Uses any global variables	-20.00
Required comments and honor statement not included at start of file exactly as specified	-35.00
Late penalty for each 15 minutes late	-2.50

The table below contains the detailed rubric of specific points for this project.

	Grade Standards - Missing: 0%, Poor: up to 50%, Fair: up to 67%, Good: up to 82%, Excellent: up to 99%, Perfect: 100%	
	Rubric	**100.00**
1	**Code Quality, Formatting, etc.**	***25.00***
	proper indentation	3.00
	good variable and constant names	2.00
	good use of constants	3.00
	good use of comments	3.00
	good use of vertical white space to separate code	2.00
	good use of horizontal white space to improve readability	2.00
	line length less than 100 characters	2.00
	accepts full path and filename as command line argument	2.00
	Creates an output file in the same directory as the input data file	2.00
	tests for file open error and gracefully exits upon error	4.00
2	**Date class**	***30.00***
	overloaded != operator accurately compares two date objects	3.00
	overloaded <= operator accurately compares two date objects	3.00
	overloaded < operator accurately compares two date objects	3.00
	overloaded >= operator accurately compares two date objects	3.00
	overloaded > operator accurately compares two date objects	3.00
	overloaded << outputs a date value in same format as input file data	4.00
	overloaded >> correctly updates data members with values extracted	4.00
	copy constructor implemented correctly and makes accurate copy	2.00
	overloaded assignment operator implemented correctly and works correctly	2.00
	overloaded assignment operator avoids self-assignment	1.00
	includes all members specified in UML diagram	2.00
3	**MArecord Class**	***25.00***
	overloaded >> operator successfully extracts one record from stream	5.00
	overloaded >> makes use of overloaded >> of Time and Date classes	4.00
	overloaded << operator successfully inserts one record to stream	5.00
	overloaded << makes use of overloaded << of Time and Date classes	4.00
	includes all data members specified in the UML diagram	2.00
	constructor uses member initialization list for composite data members	3.00
	includes all member functions specified in the UML diagram	2.00
4	**MAlog class**	***6.00***
	constructor calls private function inputRecords to process input file	2.00
	overloaded getRecordCount() functions accurately compute totals	3.00
	includes all members specified in the UML diagram	1.00
5	**MAlog inputRecords() member function**	***8.00***
	tests for file open error	2.00
	gracefully exits upon error	2.00
	successfully reads all records of the file (no more, no less)	1.00
	stores file data in appropriate object data members	1.00
	stores one MArecord object in the vector member for each input row	2.00
6	**MAlog outputRecords() member function**	**6.00**
	outputs records to the stream in the same order as the input file	2.00
	output data matches the record layout and values of the input data	2.00
	outputs the same number of records as contained in the input file	2.00

Appendix

The data for this project is an extract of a data set maintained by the Department of Labor. The complete data set contains detailed information about mining accidents and injuries since 1983. The table below lists information about the fields in the data set.

Source: http://www.data.gov/energy/datasets/accident-injuries-data-set?comm=3

COLUMN_NAME	DATA_TYPE	DATA_LENGTH	FIELD_DESCRIPTION
MINE_ID	VARCHAR2	7	Identification number assigned to the mine by MSHA. It is the mine identification number of the mine where the accident/injury/illness occurred. Use mine_id to join to the Mines and Inspections tables. Use mine_id and cal_yr to join to mine_id and cal_
CONTROLLER_ID	VARCHAR2	7	Identification number assigned by MSHA Assessments for a Legal Entity acting as a controller of an operator at the time of the accident. May not contain data (null values) if this record pertains to a contractor working at the mine.
CONTROLLER_NAME	VARCHAR2	60	Name of the controller active at the time of the accident. May not contain data (null values) if this record pertains to a contractor working at the mine.
OPERATOR_ID	VARCHAR2	7	Identification number assigned by MSHA for a Legal Entity acting as an operator at a mine at the time of the accident. May not contain data (null values) or a contractor id if the accident involved a contractor.
OPERATOR_NAME	VARCHAR2	60	Name of the operator active at the time of the accident. May not contain data (null values) or a contractor name if the accident involved a contractor.
CONTRACTOR_ID	VARCHAR2	7	Identification number assigned by MSHA for companies working as contractors at a mine. It is the contractor id of the contractor or contractor employee involved in the accident/injury. May not contain data (null values) if the accident involved an opera
DOCUMENT_NO	VARCHAR2	12	Document number assigned to the accident/injury form. The first number designates the type of document: (1) employment, (2) injury and (3) closing document. The next seven are the Julian date (YYYYDDD). The last four characters are a sequential number
SUBUNIT_CD	VARCHAR2	2	Code that identifies the location within a mine where the accident/injury/illness occurred.
SUBUNIT	VARCHAR2	32	Description of the subunit code referring to the location within a mine where the accident/injury/illness occurred: (01) Underground operation; (02) Surface operation at underground mine; (03) Strip, quarry or open pit; (04) Auger (Coal only); (05) C
ACCIDENT_DT	DATE	10	Date the accident/injury/illness occurred (mm/dd/yyyy).
CAL_YR	NUMBER	4	Calendar Year in which the accident/injury/illness occurred. It is based on the accident date.
CAL_QTR	NUMBER	1	Calendar Quarter in which the accident/injury/illness occurred. It is based on the accident date.

FISCAL_YR	NUMBER	4	Fiscal Year in which the accident/injury/illness occurred. MSHA's fiscal year begins October 1 and ends September 30. It is based on the accident date.
FISCAL_QTR	NUMBER	1	Fiscal Quarter in which the accident/injury/illness occurred. It is based on the accident date.
ACCIDENT_TIME	NUMBER	4	Time the accident/injury/illness occurred (24-hour clock).
DEGREE_INJURY_CD	VARCHAR2	2	Code identifying the degree of injury/illness to the individual. 01 Cases resulting in death; 02 Cases resulting in permanent disability; 03 Cases resulting in days away from work only; 04 Cases resulting in days away from work and days of restricted duty; 05 Cases resulting in days of restricted duty only; 06 Cases resulting in medical treatment but no days away from work or restricted duty; 07 Cases of diagnosed occupational illness; 08 Cases of fatal or non-fatal injuries due to natural causes of employees on mine property; 09 Cases of fatal or non-fatal injuries to non-employees on or off mine property; 10 All other cases.
DEGREE_INJURY	VARCHAR2	30	Description of the degree of injury/illness to the individual: (00) Accident only; (01) Fatality; (02) Permanent total or permanent partial disability; (03) Days away from work only; (04) Days away from work and restricted activity; (05) Days restri
FIPS_STATE_CD	VARCHAR2	2	Federal Information Processing Standard (FIPS) state code. It refers to the state in which accident/injury/illness occurred.
UG_LOCATION_CD	VARCHAR2	2	Code identifying the underground location where the accident/injury/illness occurred.
UG_LOCATION	VARCHAR2	28	Description of the underground location code where the accident/injury/illness occurred: (01) Vertical shaft; (02) Slope/inclined shaft; (03) Face; (04) Intersection; (05) Underground shop/office; (06) Last open crosscut; (07) Inby permanent suppor
UG_MINING_METHOD_CD	VARCHAR2	2	Description of the underground mining method code where the accident/injury/illness occurred.
UG_MINING_METHOD	VARCHAR2	20	Description of the underground mining method code where the accident/injury/illness occurred: (01) Longwall; (02) Shortwall; (03) Conventional Stoping; (05) Continuous Miner; (06) Hand; (07) Caving; (8) Other; (?) No Value Found.
MINING_EQUIP_CD	VARCHAR2	6	Code to designate the type of equipment involved in the incident. May be '?' if invalid code.
MINING_EQUIP	VARCHAR2	56	Description for the type of mining equipment involved in the accident. May be 'No Value Found' if invalid mining equipment code.
EQUIP_MFR_CD	VARCHAR2	4	Codes that identify the manufacturer of equipment involved in the incident. May be '?' if invalid code.
EQUIP_MFR_NAME	VARCHAR2	50	Mining machine manufacturer of a machine involved in an accident. May be 'No Value Found' if invalid equipment manufacture code.
EQUIP_MODEL_NO	VARCHAR2	25	Model number that identifies the equipment involved in the incident.
SHIFT_BEGIN_TIME	NUMBER	4	Time the shift started (24-hour clock) during which the incident occurred.

CLASSIFICATION_CD	VARCHAR2	2	Accident or injury classification code. The accident classification identifies the circumstances which contributed most directly to the resulting accident.
CLASSIFICATION	VARCHAR2	37	Description of the accident classification code that identifies the circumstances which contributed most directly to the resulting accident.
ACCIDENT_TYPE_CD	VARCHAR2	2	The accident type code identifies the event which directly resulted in the reported injury/accident.
ACCIDENT_TYPE	VARCHAR2	30	Description of the accident type code.
NO_INJURIES	NUMBER	4	Number of reportable injuries or illnesses resulting from the accident. Zero is a valid number if no employees were injured in the accident or if the injury was not reportable.
TOT_EXPER	NUMBER	4,2	Total mining experience of the person affected calculated in decimal years. The calculation uses both the years and months experience. May not contain data (null values).
MINE_EXPER	NUMBER	4,2	Total experience at a specific mine of the person affected calculated in decimal years. The calculation uses both the years and months experience at the mine. May not contain data (null values).
JOB_EXPER	NUMBER	4,2	Experience in the job title of the person affected calculated in decimal year. The calculation uses both the years and months experience. May not contain data (null values).
OCCUPATION_CD	VARCHAR2	3	Three-digit occupation code of the accident victim's regular job title.
OCCUPATION	VARCHAR2	40	Description of the occupation code.
ACTIVITY_CD	VARCHAR2	3	Code for the specific activity the accident victim was performing at the time of the incident. May be '?' if invalid code.
ACTIVITY	VARCHAR2	33	Description of the activity code. May be 'No Value Found' if activity code is invalid.
INJURY_SOURCE_CD	VARCHAR2	3	Code identifying the source of the injury or illness. The source of injury identifies the object, substances, exposure or bodily motion which directly produced or inflicted the injury. May be '?' if invalid code.
INJURY_SOURCE	VARCHAR2	25	Description of the injury source code. May have a description of 'No Value Found' if invalid injury source code.
NATURE_INJURY_CD	VARCHAR2	3	The nature of injury identifies the injury in terms of its principle physical characteristics. May be '?' if invalid code.
NATURE_INJURY	VARCHAR2	25	Description of the nature of the victim's injury. May have a description of 'No Value Found' if invalid nature of injury code.
INJ_BODY_PART_CD	VARCHAR2	3	Code identifying the part of the body affected by an injury. May be '?' if invalid code.
INJ_BODY_PART	VARCHAR2	39	Description of the part of the body affected by an injury. May have a description of 'No Value Found' if invalid injured body part code.

SCHEDULE_CHARGE	NUMBER	4	Charge in days lost for any permanent injury/illness. Example: 6000 days for a death, 2400 days for the loss of a foot at the ankle. May not contain data (null values) if situation does not apply.
DAYS_RESTRICT	NUMBER	4	Number of days of restricted work activity due to the injury/illness. May not contain data (null values) if situation does not apply.
DAYS_LOST	NUMBER	3	Actual days lost from work due to the injury/illness. May not contain data (null values) if situation does not apply.
TRANS_TERM	VARCHAR2	1	Indicates if the injured/ill employee was permanently transferred or terminated (Y or N). May not contain data (null values) if situation does not apply.
RETURN_TO_WORK_DT	DATE	10	Date the injured/ill employee returned to work (mm/dd/yyyy). May not contain data (null values) if situation does not apply.
IMMED_NOTIFY_CD	VARCHAR2	2	Code describing if the accident is one of the 12 types immediately reportable to MSHA.
IMMED_NOTIFY	VARCHAR2	20	Description of the Immediate Notification code: (01) Death; (02) Serious injury; (03) Entrapment; (04) Inundation; (05) Gas of dust ignition; (06) Mine fire; (07) Explosives; (08) Roof fall; (09) Outburst; (10) Impounding dam; (11) Hoisting; (12) Offsite
INVEST_BEGIN_DT	DATE	10	Date the MSHA accident investigation started (mm/dd/yyyy).
NARRATIVE	VARCHAR2	384	Narrative description of the accident/injury/illness.
CLOSED_DOC_NO	VARCHAR2	12	Document number assigned to the Returned to Duty form. The first character is designated as a '3' for the closing document. The next seven are the Julian date (YYYYDDD). The last four characters are a sequential number. May not contain data (null value
COAL_METAL_IND	VARCHAR2	1	Identifies if the accident occurred at a Coal or Metal/Non-Metal mine.