

Semester

-semesterName:string

-semesterStartDate:Date

-semesterEndDate:Date

friend ostream& operator<<(ostream& , const Semester&)// not a class memeber!

friend istream& operator>>(istream& , Semester&)//not a class member!

+Semester(string = "VACATION", Date = Date(), Date = Date())

+Semester& setSemesterName(string)

+Semester& setSemesterStartDate(Date)

+Semester& setSemesterEndDate(Date)

+string getSemesterName()

+Date getSemesterStartDate()

+Date getSemesterEndDate()

CourseSchedule

-studentName:string

-semester:Semester

-*coursesList:Course

-maxSize:int

-numCourses:int

-bool checkDates(Semester , Date , Date)

struct NoCopy // dissallow copying

{

NoCopy & operator =(const NoCopy &) = delete;

NoCopy (const NoCopy &) = delete;

}// from the new standard, didn't test it yet.

friend ostream& operatr<<(ostream& , const CourseSchedule&) // not member!

+CourseSchedule(string , Semester , int = 0)

~CourseSchedule()

+string getStudentName()

+Semester getSemester()

+int getNumCourses()

+void setStudentName(string)

+void AddCourse(Course)

+void removeCourse(Course) // **the idea is we need to remove an element from an array. I remeber doing something similiar in JAVA. I'll try to figure it out. any suggestions tho?**

Time

-hour:int

-minute:int

-standardToUniversal():void

friend ostream& operator<<(ostream& , const Time&)

//not a class member!

friend istream& operator>>(istream& , Time&) // not a class member!

+Time(int = 0, int = 0)

+Time& setTime(int, int)

+Time& setHour(int)

+Time& setMinute(int)

+int getHour() const

+int getMinute() const

+void printUniversal() const

+void printStandard() const //use the overload<< instead

+Double operator-(const Time&)

Course

-courseNumber:string

-courseName:string

-courseMeetingDays:string

-numUnits:double

-courseStartDate:Date

-courseEndDate:Date

-startTime:Time

-endTime:Time

friend ostream& operator<<(ostram& , const Course&) // not a class member

+Course(string = "EMPTY", string = " ", string " ", double = 0.0, Date = Date(), Date = Date(), Time = Time(), Time = Time())

+~Course()

+string getCourseNumber() const

+string getCourseName() const

+string getCourseMeetingDays() const

+string getNumUnits() const

+Date getCourseStartDate() const

+Date getCourseEndDate() const

+Time getStartTime() const

+Time getEndTime() const

+Course& setCourseNumber(string)

+Course& setCourseName(string)

+Course& setCourseMeetingDays(string)

+Course& setNumUnits(double)

+Course& setCourseStartDate(Date)

+Course& setCourseEndDate(Date)

+Course& setCourseStartTime(Time)

+Course& setEndTime(Time)

+Double calcDuration(const Time&)//use the overloaded operator from Time

Date

-month: int

-day: int

-year: int

friend Date& operator<<(ostream& , const Date&) //not a class member!

freind Date% operator>>(istream& , Date&) // not a class member!

+Date(int = 1, int = 1, int = 1990)

+ate setDate(int, int, int)

+Date& setMonth(int)

+Date& setDay(int)

+Date& setYeat(int)

+int getMonth()

+int getDay()

+int getYear()

+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