

CRC CARDS – HAUNTED PYRAMID

[HTTPS://GITHUB.COM/CSC207-UOFT/COURSE-PROJECT-HAUNTED-PYRAMID](https://github.com/CSC207-UOFT/COURSE-PROJECT-HAUNTED-PYRAMID)

OCTOBER 15TH, 2021

SUMMARY

- Entities
 - User
 - Student
 - Event
 - Course
 - Assignment
 - OurCalendar
- To be implemented
 - Lecture
 - Reading
 - Test
 - Tutorial
 - Labs
 - Others
- Use Cases
 - CalendarManager
 - CourseManager
 - EventManager
 - StudentManager
- Controllers
 - CourseController
 - EventController
 - LoginController
 - MainController
 - StudentController
- Interface
 - Fluid
 - Repeated
- Presenter
 - CalendarPresenter
- Gateway
 - IOSerializable

ENTITY User

Student

Responsibilities	Collaborators
<ul style="list-style-type: none">• Id• Name• Username• Password• birthDate• dateCreated• emailAddress• phoneNumber• homeAddress	
<ul style="list-style-type: none">• Setter of the above responsibilities	
<ul style="list-style-type: none">• Getter of the above responsibilities	
<ul style="list-style-type: none">• Construct with name and birthDate	

ENTITY

Student

Responsibilities	Collaborators
<ul style="list-style-type: none">• number of courses• courses enrolled	<ul style="list-style-type: none">• Course
<ul style="list-style-type: none">• AddCourse• RemoveCourse• Getter methods	

ABSTRACT

ENTITY Event

Reading, Assignment, Lecture,
Lab, Tutorial, Test, Other

Responsibilities	Collaborators
<ul style="list-style-type: none">namelocation (Optional)startTimeendTime	
<ul style="list-style-type: none">modify startTime and endTime	
<ul style="list-style-type: none">Setters for startTime, endTime, name and location	
<ul style="list-style-type: none">Getters for startTime, endTime and length	
<ul style="list-style-type: none">HasConflict	
<ul style="list-style-type: none">Equals	
<ul style="list-style-type: none">ToString	

ENTITY
Assignment

Responsibilities	Collaborators
<ul style="list-style-type: none">• worksessionsLength	<ul style="list-style-type: none">• Fluid
<ul style="list-style-type: none">• hoursSpent	
<ul style="list-style-type: none">• hoursNeeded	
<ul style="list-style-type: none">• DueDate	

ENTITY

Course

Responsibilities	Collaborators
<ul style="list-style-type: none">• name• EventList [events share the same course attribute organized in a map based on event type]	<ul style="list-style-type: none">• Event• Lecture• Assignment• Reading• Test• Tutorial• Labs• Others
<ul style="list-style-type: none">• Add events to task	
<ul style="list-style-type: none">• Remove events from task	
<ul style="list-style-type: none">• Getter of name and task	

ENTITY

OurCalendar

Responsibilities	Collaborators
<ul style="list-style-type: none">• calendarMap• conflict (Boolean)• conflictEvent• dateInfo• calendarMap	<ul style="list-style-type: none">• Event• Lecture• Assignment• Reading• Test• Tutorial• Labs• Others
<ul style="list-style-type: none">• add and remove objects from calendarMap	
<ul style="list-style-type: none">• check conflicts on calendarMap	
<ul style="list-style-type: none">• Getter of conflictEvent, conflict, dateInfo and calendarMap	

USE CASE CalendarManager

Responsibilities	Collaborators
<ul style="list-style-type: none"> • CurrentManager • futureManager • pastCalendar • currentYear, currentMonth, currentDate 	<ul style="list-style-type: none"> • OurCalendar • Lecture • Assignment • Reading • Test • Tutorial • Labs • Others • Event
<ul style="list-style-type: none"> • Generate and store past and future calendars 	
<ul style="list-style-type: none"> • Getters for Monthly Calendar, Weekly Calendar and Daily Calendar, currentYear, currentMonth, currentDate 	
<ul style="list-style-type: none"> • Add and remove events to/from a Calendar 	
<ul style="list-style-type: none"> • Notify conflict events for chosen date • Get event time and date info for the selected date 	

USE CASE EventManager

Responsibilities	Collaborators
<ul style="list-style-type: none">• Map of Events• Map of Unscheduled Hours• Map of Scheduled Hours• getDay• getType• GetEvent• RemoveEvent• Rescheduable• addEvent• replaceEvent• totalHours• freeTime• chronological	<ul style="list-style-type: none">• CalendarManager• Event• Fluid• Repeated

USE CASE

StudentManager

Responsibilities	Collaborators
<ul style="list-style-type: none">• Student id to student courses map• Student id to student object map• Create a studentMap with given information and acquired courseList from CourseManager.• Store student information in studentInfo• Check username and password• getStudentInfo• getStudentMap• addNewStudent	<ul style="list-style-type: none">• Student• Course

USE CASE

CourseManager

Responsibilities	Collaborators
<ul style="list-style-type: none">• courseList (array of Course's that the events created belong to)• Create courseList of the current user.• Getter of courseList• Getter of one specific course information from courseList• Add Course to courseList• Remove Course from courseList	<ul style="list-style-type: none">• Course• Event• Student

CONTROLLER

MainController

Responsibilities	Collaborators
<ul style="list-style-type: none">Instantiates IOSerializableInstantiates StudentController and LoginController and initially checks the verification of the user through log in phaseInstantiates EventControllerInstantiates CalendarPresenterDisplayInitScreenDisplayCalendargenerateEvent	<ul style="list-style-type: none">*CalendarManager*EventManagerLoginControllerStudentControllerEventControllerCalendarPresenterIOSerializable
<ul style="list-style-type: none">saveAndExitProgram	<p>*Managers will move into their own controllers after phase 0 for better organization and simplicity in the MainController</p>

CONTROLLER

LoginController

Responsibilities	Collaborators
<ul style="list-style-type: none">• login• logout• signup• isLoggedIn	<ul style="list-style-type: none">• StudentController

CONTROLLERS

EventController

Responsibilities	Collaborators
Instantiate EventManager and CalendarManager	<ul style="list-style-type: none">• CalendarManager• EventManager• CalendarPresenter
Schedule – take user input to schedule, modify, remove events	

CONTROLLERS

StudentController

Responsibilities	Collaborators
<ul style="list-style-type: none">Initialize StudentManager based on readability of the serializable file. If there isn't a saved data in the directory, create a sample file: "students.ser" with sample login information. If there is a file, read from it and instantiate StudentManager with the arraylist of students obtained from the serializable file.	<ul style="list-style-type: none">IOSerializableStudentManager

Fluid

Responsibilities	Collaborators
<ul style="list-style-type: none">• Setter for fluid hours remaining	<ul style="list-style-type: none">• AutoSchedule
<ul style="list-style-type: none">• Getter for fluid hours remaining	
<ul style="list-style-type: none">• Setter for work session length	
<ul style="list-style-type: none">• Getter for work session length	
<ul style="list-style-type: none">• Getter for session	
<ul style="list-style-type: none">• scheduleSession	

AutoSchedule

Responsibilities	Collaborators
<ul style="list-style-type: none">• Setter for fluid hours remaining	

Repeated

Responsibilities	Collaborators
<ul style="list-style-type: none">• Setter for repitition	
<ul style="list-style-type: none">• Getter for list and start times	
<ul style="list-style-type: none">• @Override conflicts	

IOSerializable

Responsibilities	Collaborators
<ul style="list-style-type: none">• Logger – logs any errors or exceptions• EVENTS_FILEPATH• TASKS_FILEPATH• STUDENTS_FILEPATH• CALENDAR_FILEPATH• hasSavedData - checks if there is any saved data in the directory and return a boolean	<p>**Student – this is necessary because we need to type cast the read object as an <code>ArrayList<Student></code> when deserializing the file. This could change through code optimization or new insight for later phases of the program.</p>
<ul style="list-style-type: none">• studentsReadFromSerializable – read from "students.ser" and return an ArrayList of Students that were saved in the directory• studentsWriteToSerializable – write to "students.ser" and save the ArrayList of students	
<ul style="list-style-type: none">• ***It is important to note that for phase 0, serialization of students were only implemented fully. Going forward, most if not all entities will follow	

PRESENTER

CalendarPresenter

Responsibilities	Collaborators
<ul style="list-style-type: none">Starts from an empty calendar	<ul style="list-style-type: none">Calendar Manager
<ul style="list-style-type: none">Add information from the calendar manager	
<ul style="list-style-type: none">Display information as string from calendar manager	