MATH DEPARTMENT SCHEDULER

VERSION: 0.0000000001

DATE: FEBRUARY 18, 2014

DEVELOPERS:

Saul Castro

Shin Kang

Ving Trung

Yekun Yang

Prepared for Pasadena City College Math Department

Spring 2014

**1. Introduction**

The program is designed to assist the Dean of Math Department to organize and determine the course schedule based on the survey sheets done by the instructors, including the general information of the instructors, 1st/2nd session option, number of classes per session, available class time, and course options, etc.

## 1.1 Purpose

The program will generate an Excel spreadsheet containing all the options from instructors. The final decision of course schedule will be made by the Dean of Math Department. However the output result of the program will provide several friendly options for the Dean, including drop-down menus for sorting, empty/contradicted time spot, and different priorities of the instructors.

**2. Specific Requirements**

1) Gather faculties informations.

a) Last and first name

b) Faculty ID number (8 digits)

c) Requests

-Amount of sessions desired? (first, second)

-How many classes per session? (one, two)

-8 week class? (ONLY FOR 2nd session)

-Time desired for first session? (predefined)

-Time desired for second session? (predefined)

-Courses desired? (predefined)

-Room preference?

1. Board (white, black, smartboards, neither).

2) Display Instructor’s name, course name, time requested, and full-time or adjunct determinant on an excel sheet sorted by course name.

3) Determine if faculty is full time or adjunct.

a) Sort excel document to display a list of full time instructor and a list of adjunct.

**3. Class and Objects TOP DOWN**

**Input Document**

**Input File:**

CSV file containing class request:

(example)

|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor name** | **Course Name** | **Time Requested** | **Full Time or Adjunct** |
| instructor 1 | Math 101 | 3:00pm - 4:00pm | Full Time |
| instructor 2 | CS 101 | 3:00pm - 4:00pm | Adjunct |
| instructor 2 | Math 101 | 4:00pm - 5:00pm | Adjunct |
| instructor 3 | Math 101 | 3:00pm - 4:00pm | Full Time |

1. Use grab\_instructor() to create new Instructor class for each instructor

***Class: Instructor***

**String** firstName //first name

**String** lastName //last name

**Int** idNum //faculty ID number, 8-digits long

**Boolean** Adjunct // True or False

Method

Public String name(){ return firstName + “ “ + lastName; }

Public String adjunct(){ if(Adjunct) return “Adjunct” else return “Full Time”;}

2. Use grab\_course to create Course class for each course

***Class: Course***

**String** CourseName:

**Int** ClassNumber:

Method

Public String name(){return CourseName;}

3. Create TimeSlot Class for each time slot

**Class: TimeSlot**  
**Arraylist<Int>** TimeSlot :

*Methods*

public int stringTimeConverter() // changes the string time to an integer military time

4. Add Request class for each request

**Class: Request (Instructor, Course, Time)  
Instructor** InstructorObject:

**Course** CourseObject:  
**Time** TimeObject:

Methods

**Public ArrayList(Object)** Sort\_Request\_Array();

5. Use interface to sort request Objects

**Method:**

**SortRequest(ArrayList(Objects))** // sort the request

6. Print out the csv file using a loop through the output of the scheduler;

**Class PrintSchedule(ArrayList(Object))** //print out request object in csv format

**Classes**

***Class: Instructor***

**String** firstName //first name

**String** lastName //last name

**Int** idNum //faculty ID number

**Boolean** Adjunct // True or False

Method

Public String name(){ return firstName + “ “ + lastName; }

Public String adjunct(){ if(Adjunct) return “Adjunct” else return “Full Time”;}

***Class: Course***

**String** CourseName:

**Int** ClassNumber:

Method

Public String name(){return CourseName;}

**Class: TimeSlot**  
**String** TimeSlot:

**Class: Request (Instructor, Course, Time) implements Scheduler  
Instructor** InstructorObject:

**Course** CourseObject:  
**Time** TimeObject:

*Methods*

**Public ArrayList(Object)** Sort\_Request\_Array() //

**Scheduler(interface)**

**Public ArrayList(Object)** Sort\_Request\_Array(Request Object)() // sort the request

**Class PrintSchedule(ArrayList(Object))** //print out request object in csv format

**Methods**  
Public void grab\_instructor(Scanner in) //input scanner

//create instructor class objects

Public void grab\_course(Scanner in) //input scanner

//create course class objects

**Output:**

CSV file

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Math 101** | **Instructor name** | **Course Name** | **Time Requested** | **Full Time or Adjunct** |
|  | instructor 1 | Math 101 | 3:00 - 4:00 | Full Time |
|  | instructor 3 | Math 101 | 3:00 - 4:00 | Full Time |
|  | instructor 2 | Math 101 | 4:00 - 5:00 | Adjunct |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CS 101** | **Instructor name** | **Course Name** | **Time Requested** | **Full Time or Adjunct** |
|  | instructor 2 | CS 101 | 3:00 - 4:00 | Adjunct |