Miami University Class Schedule Validator

Adam Brush, Tony Ciraci, Yan Yu

October, 2013

I. Abstract

This document discusses the Miami University Class Schedule Validator. The Miami University Class Schedule (M.U.C.S) Validator is designed for use in individual departments, divisions, and the University Registrar's Office of Miami University to help monitor and comply with a set of class scheduling parameters and rules set by the Miami University Office of the Provost, defined as the Oxford Class Schedule Policy (Policy). (Miami University, 2011). The application analyzes proposed schedules and checks them for validation by generating information showing the compliance of schedules with the Policy parameters and rules.

II. Background for The Project

Each semester, the departments and programs of Miami University create a schedule for their classes for the following semester, deciding when their classes will be for both time and day. In the past, classes have been scheduled more heavily during certain days and times, other days and times are underutilized; increasingly more classes are being scheduled outside of the established timeblocks, resulting in suboptimal use of classrooms, timeblocks, and availability of classes for students.

To address this problem, the Office of Enrollment Management of the university has a set of scheduling policies and rules dictating what day and what timeblocks are allowed for departments to schedule classes (Miami University, 2011). The policies are shown in Appendices I and II. The goal of the policies and rules is to spread classes out across days and times evenly so as to maximize student access to classes to insure student success and to make the best use of resources including faculty, classrooms, and other facilities.

The University Registrar's Office is tasked with validating compliance with rules. Currently, departments and programs submit schedules to the Registrar, and compliance is checked by hand to ensure that the policy rules are met. The University Registrar's Office needs an application to check compliance that could be used by departments before schedules are submitted that office. This application would compare draft schedules to the scheduling rules and generate reports showing how well the schedule complies with the rules. This application would be able to be distributed to the heads of all departments and programs for use when creating their schedules. The University Registrar's Office would also use the application to monitor compliance by departments, divisions, and across the entire university.

III. Project Goals

The following are the desired goals of the project:

- Develop an application to view schedule compliance with scheduling rules
- Distribute the application to all departments to validate the draft schedules before submitting these schedules to Registrar
- Distribute the application to Registrar to validate the submitted schedules

- Help each department to create a classes schedule meeting the scheduling rules
- Help arrange classrooms and time more efficiently

IV. The Database

When departments create a class schedule that information is stored in the University's central Oracle database in the form of tables (Oracle, n.d.). This project is linked to the live Oracle schedule database. For the Validator application, we created a login interface, so people who will use this application can input their database accounts' information to connect to the University's central Oracle database. This section describes the tables we used for this application.

1. Tables provided by Registrar

The following tables were provided by the Registrar. Those marked with * are used by the schedule validator.

- **SCBCRKY**
  This table contains data about each class. Some of the data includes when the class was created, what term it was first offered, what term the class was last offered or if the class is still being offered

- **SCBCRSE***
  This table contains data about each class and specifically which division, department and subject area that it belongs to. For each class listed, the effective term is given

- **SIRASGN**
  This table contains data about faculty member institutional assignments. This includes the term of the class as well as the related crn number

- **SOBPTRM**
  This table contains data about section part-of-term codes. The term code is related to each of the different types of classes. Some examples would be full length semester, first half sprint, and second half sprint. The corresponding start and end dates for each one of these types of class terms is listed

- **SSBSECT***
  This table contains data about general section information for each class. This includes the current term that the class is offered, the crn, which part-of-term it belongs to, class number, and subject code. This table holds all the base information for each section

- **SSRATTR***
  This table contains data about each crn and which class attributes that specific class is a part of. The corresponding term code for each class is given

- **SSRMEET***
  This table contains data about all of the section meeting times. For each class, the building code, building room number, start time, end time, start date, end date, and weekly meeting patterns are displayed

- **STVCAMP***
  This table contains data about campus validation. It links each campus to its unique campus code as well as when the starting date of that particular campus

- **STVCOLL***
  This table contains data that links each divisional code to the corresponding division. This table is used to display the full name of the divisions in the drop-down lists of the program

- **STVDEPT***
  This table contains data that links each department code to the corresponding department. This table is used to display the full name of the departments as well as the populating the drop-down lists in the department selection list

- **STVGMOD**
  This table contains data about grading codes. It links each grading code to the corresponding grading mode (e.g. pass/fail, audit, normal grading)

- **STVMEET***
  This table contains data about all of the valid day of week and time combinations. There are currently 618 valid meeting patterns

- **STVPTRM**
  This table contains data about part-of-term validation. It links each part-of-term code to its corresponding part-of-term description

- **STVSAPR**
  This table contains data special approval validation. Each special approval code is linked to the corresponding special approval description

- **STVSCHD**
  This table contains data about schedule type code validation. This links each schedule code to its full length description

- **STVSSTS**
  This table contains data about section status validation. Each section status code is linked to its corresponding description. A few examples include active, inactive, and cancelled

- **STVSUBJ**
  This table contains data about subject validation. It links each subject code to its full length name of the subject code

V. Software Architecture & Implementation

The application is implemented using the Java programming language, along with the self-contained SQLite database (Oracle-Java, n.d.).

1. General Description of the Process

The program starts with the *UserInterface* class which creates the Graphical User Interface (GUI) and allows the user to interact with the program and the database. When the user makes selections in the GUI, it creates a *ConsoleConnection* object which takes care of most of the queries that are run in the database. The *ConsoleConnection* is what is used to obtain the list of courses based on what specifications the user selected in the GUI. For example if the user selected all English 111 courses and clicks “calculate”, the *ConsoleConnection* would return an arraylist of all of the English 111 courses regardless of whether they passed or failed.

From there this arraylist of courses is passed into the *Validator* class which is the part of the code that actually goes through and checks each class section to see whether it was in one of the approved timeblocks. Essentially, the Validator separates the original arraylist of courses into two new arraylists, one for all of the passed courses and one for all of the failed courses. The Validator then passes these two arrays into the *StatGenerator* which is responsible for populating most of the data that you see when you run the calculate. From there, the *StatGenerator* will use the passed and failed arrays to calculate all of the percentages and counts that you see in the GUI.

2. Flow Chart & Major Modules of the Program

1.1 Flow Chart

Figure 2.1 is the flow chart of the schedule validator, which depicts the general working process and data flow of the application.

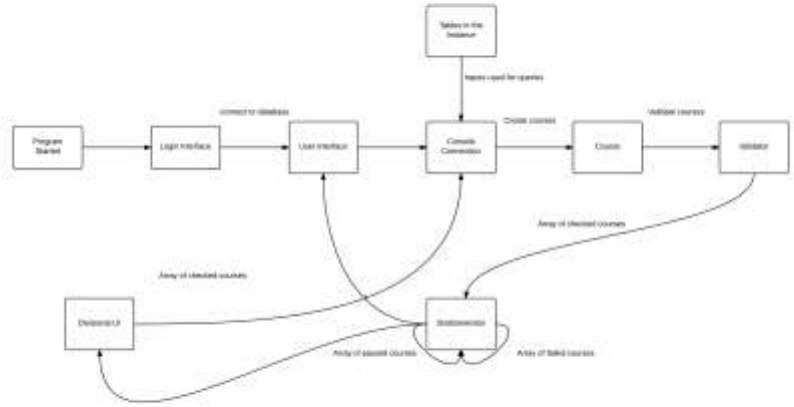


Figure 2.1 Major Software Components and Data Flow

1.2 Major Modules

1.2.1 UserInterface

This class is in charge of creating the user interface that is used to interact with the database of courses. Once this class is run, the program is running. The program will not run from any of the other classes. Specifically, this class allows the users to select which set of conditions (in the dropdowns) they want to validate however it does not do any of the calculations or queries. Its only focus is on updating the UI when a user makes selections within the dropdowns provided to them and clicks the 'Calculate' button. Based on the dropdowns selected this class uses a ConsoleConnection to create the arraylist of courses to be used to populate the UI. From there it creates a Validator to validate the courses and a StatGenerator to calculate all stats to be displayed in the UI. The SchoolReportUI class does the same thing as UserInterface except that it does so for entire divisions.

1.2.2 ConsoleConnection

This class is in charge of creating and the connection to the database as well as running any of the queries needed for the program. The UserInterface provides the data from the dropdowns selected by the user and the ConsoleConnection then goes and queries the database for the resulting courses. It is the job of the ConsoleConnection to actually create the arraylist of courses that will be validated. It does not do the actual validation of the courses. ConsoleConnection is also responsible for running all of the queries

that help populate the dropdowns as the user selects them.

1.2.3 Validator

The Validator class is in charge of actually validating the courses that are returned from the ConsoleConnection. It uses the arraylist of courses as well as its own private connection to the database to check the times of each of the courses against the registrars policy of acceptable start and end times for courses (see Appendix II), which is located in a table in the database. Lastly, the validator is in charge of assigning a pass or fail variable to each of the courses. This pass and fail variable is important when it comes time for the newly transformed arraylist of validated courses to be passed on to the StatGenerator class.

1.2.4 StatGenerator

The StatGenerator is responsible for taking an arraylist of validated courses and calculating all of the statistics to be displayed in the StatGenerator. Firstly, before any stats are calculated, the StatGenerator takes the arraylist of validated courses and splits it into two separate arraylists, one for passing and one for failing courses. From there, all other methods in the StatGenerator use these two arraylists to do all of the calculations needed for the program. The actual StatGenerator method calls themselves are done in the UserInterface after the user has decided on a set of courses they want to validate.

2. Assumptions in the Program

The following schedule codes are not included in the data pulled from the database:

- Courses listing an exam time are included but the listing for their corresponding exam date(s) is not included in the data.
- Sprint Courses are not included in the data. Only full term courses are listed in the data.
- Online courses are not included in the data
- Although ‘Block 1’ lists 8:30 to 9:50, the algorithm for the corresponding passed for failed accepts any time between 7:00 and 9:50 as passing lists it in ‘Block 1’.

VI. Major User Interfaces

This section presents an overview of the user interface for the application. The application allows the user to validate schedules by campus, division, department, subject code, class number, and range of class numbers. Reports show summary statistics as well as details for individual class sections.

1. Main User Interface

When user first starts the application, the main user interface (UI) illustrated in Figure

6.1 will pop up.

Figure 6.1 Main User Interface

2. Class Search Criteria and Results after Validating Schedule

The user next selects search criteria for classes to be validated. Criteria can include the term, campus, division (college), department and/or a specific class number to be validated. The user may validate classes on an entire campus, division, or classes. Another criteria is a range of courses numbers, for example all courses less than or equal to ACC 221. When user validates a specific class (say ACC 221) or validates a class range, interface is update do show the number and percentage of class that pass the validation rules (Appendix I), as well as the distribution of courses in the designated time blocks (Appendix II) as follows:

Figure 6.2 Search Criteria and Validation Results

3. Detailed Class Breakdown for Passed and Failed Times

3.1 Show Passed

When user clicks show passed button (see Figure 6.2) a detailed list of passed classes is displayed as shown in Figure 6.3.

Figure 6.3 Detailed List of Passed Class Times

3.2 Show Failed

The Show Failed button will show a list like that for the passed-class list, which is the same as **Show Passed**, the difference is that the *Course* column is red.

4. Validating all Classes for a Division (College or School)

The Run Division Report button (see Figure 6.2) is a quick way to validate all classes for a division. Figures 6.5 and 6.6 illustrate that user interface.

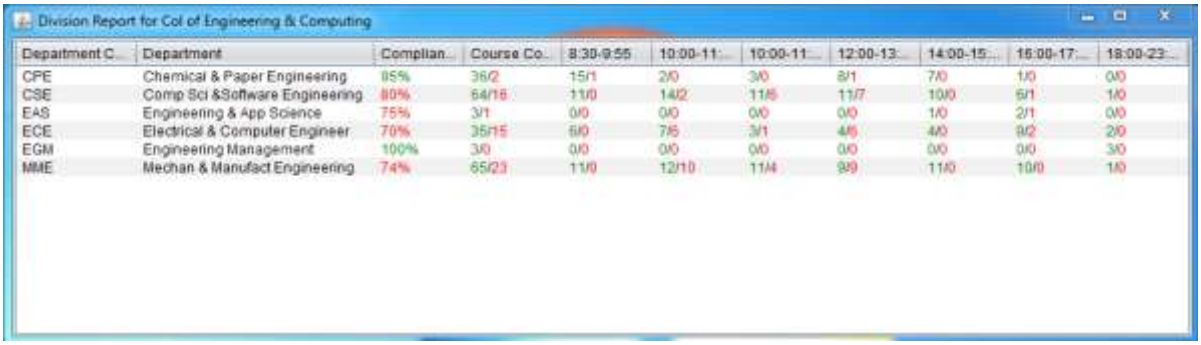
Figure 6.4 Validating an Entire Division

When user selects a division and a term(e.g. School of Engineering & Applied Science) and hits calculate button, the divisional UI will be updated as in Figure 6.5

Days Offered	%.33% of courses passed, must be above 90%.	206 courses passed out of a total of 263 courses.	Block 1 8:00 - 9:50	Block 2 10:00 - 11:20	Block 3 11:30 - 12:50	Block 4 13:00 - 14:20	Block 5 14:30 - 15:50	Block 6 16:00 - 17:50	Block 7 18:00 - 23:00
M and/or W and/or F	60.08%	141/17	27/0	25/5	19/3	21/9	26/0	19/0	4/0
T and/or R	39.92%	65/40	16/1	10/13	9/8	11/14	7/0	9/4	3/0
4 or 5 Days a Week	0%	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
Totals	%	206/57	43/1	35/18	28/11	32/23	33/0	28/4	7/0

Figure 6.5 Validation Results for School of Applied Science and Engineering

Also a detail division report will display statistics for all departments within the selected division, as shown in Figure 6.7.



Department C...	Department	Complian...	Course Co...	8:30-9:55	10:00-11:...	10:00-11:...	12:00-13:...	14:00-15:...	16:00-17:...	18:00-23:...
CPE	Chemical & Paper Engineering	85%	36/2	15/1	2/0	3/0	8/1	7/0	1/0	0/0
CSE	Comp Sci &Software Engineering	80%	64/16	11/0	14/2	11/6	11/7	10/0	6/1	1/0
EAS	Engineering & App Science	75%	3/1	0/0	0/0	0/0	0/0	1/0	2/1	0/0
ECE	Electrical & Computer Engineer	70%	35/15	5/0	7/5	3/1	4/5	4/0	9/2	2/0
EGM	Engineering Management	100%	3/0	0/0	0/0	0/0	0/0	0/0	0/0	3/0
MME	Mechan & Manufact Engineering	74%	65/23	11/0	12/10	11/4	9/9	11/0	10/0	1/0

Figure 6.7 Results for all Departments within a Division

When user clicks *Help* button on home screen, the application will use the default browser to open an URL that displays the scheduling policy.

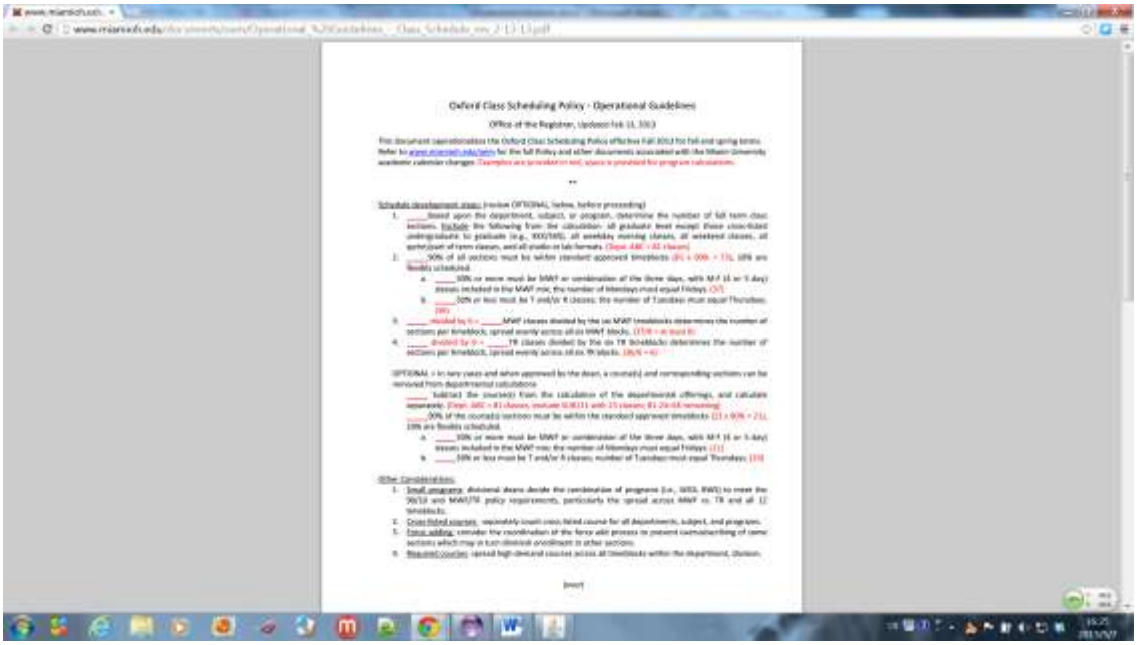


Figure 6.8 Scheduling Policy

VII. Discussion

7.1 General Overview of This Program

By using this computer program, office of registrar now can cite the fall 2012 data of timeblock spread vs fall 2013, huge differences for the good. Instead of manual review and calculation to the class schedule policy, schedule draft is calculated using validator. Departments' and divisions' leads are able to effect change in advance of regular starting. Registrar can be secondary reviewer.

As a consequence, validator saves extremely large amount of time for registrar and departments, which in turn can ensure them have plenty of time to make a good class schedule. However, validator program is finished, but it is not fully tested, there must have some bugs hidden inside of it. Thus, the results of the validating may not be 100% precise. A good test plan is currently needed in order to improve its functionalities.

7.2 Future Improvements

In this section we identify several tasks that remain for the project.

1. **Software Development Tasks:**
   - Develop a test plan to identify and document current bugs in the code.
2. **Additional Features & Functionalities to Be Added**
   - Find a way to re-arrange all the buttons and dropdown lists on the UI to make the application work under different screen resolution ratio.
   - Work with Register and IT center to find a way to convert the current app to a web-based application.

VIII. Conclusion

When the M.U.C.S. Validator is fully functioning, it will be immensely helpful to the Registrar's office and to the department chairs and program directors. Once in place, the application will make it easier to schedule classes. By the end of this semester, the Registrar will begin using the application. It will give the Registrar a more user friendly means of checking and enforcing the scheduling rules. And in order to ensure the validator get 100% precise results, a good test team and a good test plan are required in order to make sure every functionality is tested. Last, to deliver good a user experience, the UI needs to be redesigned.

Bibliography

Bravaco, Ralph & Simonson, Shai. (2009). *Java Programming: From the Ground Up.* New York, NY: McGraw-Hill.

Elmasri, Ramez A. & Navathe, Shamkant. (2010). *Fundamentals of Database System(6th Edition).* Boston, MA: Addison-Wesley.

Miami University. (2013, May 9). *Office of Enrollment Management*. Retrieved from Miami University: http://www.miamioh.edu/oem/.

Miami University. (2013, May 9). *Resource Documents*. Retrieved from Office of Enrollment Management: http://www.miamioh.edu/oem/academic-calendar/documents/index.html.

Oracle. (n.d.). *Java Home*. Retrieved from Java: http://java.com/en/.

Oracle. (n.d.). *Oracle Database.* Retrieved from Oracle: http://www.oracle.com/us/products/database/overview/index.html.

Appendix I

Oxford Class Scheduling Policy - Operational Guidelines
Office of the Registrar, Updated Feb 13, 2013

Class Schedule Development

Each departmental and program semester class schedule is to adhere to these policies as approved by the Provost; consult the document, "Approved Standard Oxford Campus Timeblocks".

Lecture classes: Percentages based on daytime course section offerings with starting times before 6:00 p.m.

- A minimum of 90% of department or program course sections must be offered within approved, standard timeblocks.
  - A minimum of 50 % of section offerings must be on the MWF timeblock model; 4 and 5 day a week classes are included in this 50% minimum.
    - To provide equal distribution, MWF, MW, MF, and WF offerings must contain the same number of classes meeting Mondays as Fridays. (ex: 10 MWF, 2 MW, 2 MF and 2 WF = 14 M and 14 F).
  - A maximum of 50 % of course section offerings can be on the TR timeblock model.
- A maximum of 10% of department or program course sections can be outside of approved, standard timeblocks:
  - All graduate level classes are excluded from the 10%
  - Course offerings with instructor types of laboratory or studio are excluded (see below)
- Course section offerings must be evenly distributed:
  - across all six daytime timeblocks MWF
  - across all six daytime timeblocks TR
- Miami Plan classes must all be offered during approved, standard timeblocks; if evening offerings, must utilize the 6:00 p.m. starting time to accommodate day classes which may not end until 5:50 p.m.

Laboratory and studio classes: Course section offerings may be offered either day or evening.

- Classes should adhere to common start times while minimizing the overlapping of approved standard timeblock.
- Laboratories and studio classes are not calculated in the 10% of department/program course sections offerings outside of approved, standard timeblocks.

Appendix II

Time Blocks – Fall 2013

Approved Standard Oxford Campus Timeblocks for Daytime Classes
To be used with the Oxford Class Scheduling Policy:
These timeblocks represent 90% of class offerings; maximum 10% of classes can be offered outside these blocks.
See Evening Class Timeblocks, page two
Office of the University Registrar (updated January 18, 2013)

* Lab/Studio class length is generally twice the number of formalized instructional minutes than required for a lecture class (2:1 ratio).
Graduate classes may be scheduled per department and student needs and may vary from these blocks.

Classes begin by 8:30 a.m. (some 8:00) and end by 5:50 p.m. - lab timeblocks occur between 8:00 a.m. and 9:50 p.m.

Time of Day	Class Block #	Monday - Wednesday - Friday MWF, MW, WF, MF, M, W, or F 55 minute classes: 1 cr = any 1 M,W,F 2 cr = any 2 M,W,F 3 cr = MWF 75 minute classes: 4 cr = MWF 80 minute classes: 3 cr = MW, MF or WF 110 minute classes: 4 cr. hr. = MW, MF or WF NOTE: Block 1 or 6 only		Tuesday - Thursday TR, T or R 55 minute classes: 1 cr = T or R 2 cr = TR 80 minute classes: 3 cr = TR 110 minute classes: 4 cr. hr. = TR NOTE: Block 1 or 6 only		Monday through Friday M through F or any 4-day combination 55 minute classes: 4 cr = any 4 days 5 cr = all 5 days 70 minute classes: 5 cr = any 4 days		* Labs (2:1 ratio) * Lab Timeblocks 110 minutes (2x55) for a one credit hour lab	Time of Day
8:00 :15 :30 :45	#1 Class Block		8:00 - 9:50 (4 cr MW, MF or WF)		8:00 - 9:50 (4 cr TR)				8:00 :15 :30 :45
9:00 :15 :30 :45		8:30 - 9:25 (1, 2, or 3 cr MWF)	8:30 - 9:45 (4 cr MWF) 8:30 - 9:50 (3 cr MW MF or WF)	8:30 - 9:50 (3 cr TR)	8:30 - 9:25 (1 or 2 cr TR)	8:30 - 9:25 (4 or 5 cr M - F)	8:30 - 9:40 (5 cr any 4 days)	* 8:00 - 9:50 #1 Lab Block	9:00 :15 :30 :45
10:00 :15 :30 :45	#2 Class Block	10:00 - 10:55 (1, 2, or 3 cr MWF)	10:00 - 11:15 (4 cr MWF) 10:00 - 11:20 (3 cr MW MF or WF)	10:00 - 11:20 (3 cr TR)	10:00 - 10:55 (1 or 2 cr TR)	10:00 - 10:55 (4 or 5 cr M - F)	10:00 - 11:10 (5 cr any 4 days)	* 10:00 - 11:50 #2 Lab Block	10:00 :15 :30 :45
11:00 :15 :30 :45	#3 Class Block	11:30 - 12:25 (1, 2, or 3 cr MWF)	11:30 - 12:45 (4 cr MWF) 11:30 - 12:50 (3 cr MW MF or WF)	11:30 - 12:50 (3 cr TR)	11:30 - 12:25 (1 or 2 cr TR)	11:30 - 12:25 (4 or 5 cr M - F)	11:30 - 12:40 (5 cr any 4 days)	* 12:00 - 1:50 #3 Lab Block	11:00 :15 :30 :45
12:00 :15 :30 :45	#4 Class Block	1:00 - 1:55 (1, 2, or 3 cr MWF)	1:00 - 2:15 (4 cr MWF) 1:00 - 2:20 (3 cr MW MF or WF)	1:00 - 2:20 (3 cr TR)	1:00 - 1:55 (1 or 2 cr TR)	1:00 - 1:55 (4 or 5 cr M - F)	1:00 - 2:10 (5 cr any 4 days)		12:00 :15 :30 :45
1:00 :15 :30 :45	#5 Class Block	2:30 - 3:25 (1, 2, or 3 cr MWF)	2:30 - 3:45 (4 cr MWF) 2:30 - 3:50 (3 cr MW MF or WF)	2:30 - 3:50 (3 cr TR)	2:30 - 3:25 (1 or 2 cr TR)	2:30 - 3:25 (4 or 5 cr M - F)	2:30 - 3:40 (5 cr any 4 days)	* 2:00 - 3:50 #4 Lab Block	1:00 :15 :30 :45
2:00 :15 :30 :45	#6 Class Block	4:00 - 4:55 (1, 2, or 3 cr MWF)	4:00 - 5:15 (4 cr MWF) 4:00 - 5:20 (3 cr MW MF or WF) 4:00 - 5:50 (4 cr MW only)	4:00 - 5:20 (3 cr TR)	4:00 - 4:55 (1 or 2 cr TR) 4:00 - 5:50 (4 cr TR)	4:00 - 4:55 (4 or 5 cr M - F)	4:00 - 5:10 (5 cr any 4 days)	* 4:00 - 5:50 #5 Lab Block	2:00 :15 :30 :45
3:00 :15 :30 :45									3:00 :15 :30 :45
4:00 :15 :30 :45									4:00 :15 :30 :45
5:00 :15 :30 :45									5:00 :15 :30 :45

Evening classes begin at 6:00 p.m.

Approved Standard Oxford Campus Timeblocks for Evening Classes

Office of the University Registrar (updated January 18, 2013)

* Lab/Studio class length is generally twice the number of formalized instructional minutes than required for a lecture class (2:1 ratio).
Graduate classes may be scheduled per department and student needs and may vary from these blocks.

Classes beginning 6:00 p.m. and after: ONE MEETING PER WEEK

	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			
	2 cr 110 min	3 cr 160 min	4 cr 215 min	2 cr 110 min	3 cr 160 min	4 cr 215 min	2 cr 110 min	3 cr 160 min	4 cr 215 min	2 cr 110 min	3 cr 160 min	4 cr 215 min	* Labs (2:1 ratio)
#7 Class Block starts 6:00 p.m.	6:00-7:50 M	6:00-8:40 M	6:00-9:35 M	6:00-7:50 T	6:00-8:40 T	6:00-9:35 T	6:00-7:50 W	6:00-8:40 W	6:00-9:35 W	6:00-7:50 R	6:00-8:40 R	6:00-9:35 R	* 6:00 - 7:50 #6 Lab Block
#8 Class Block starts 8:00 p.m.	8:00-9:50 M			8:00-9:50 T			8:00-9:50 W			8:00-9:50 R			* 8:00 - 9:50 #7 Lab Block

Classes beginning 6:00 p.m. and after: TWO MEETINGS PER WEEK

	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			
	2 cr 55 min	3 cr 80 min	4 cr 110 min	2 cr 55 min	3 cr 80 min	4 cr 110 min	2 cr 55 min	3 cr 80 min	4 cr 110 min	2 cr 55 min	3 cr 80 min	4 cr 110 min	* Labs (2:1 ratio)
#7 Class Block starts 6:00 p.m.	6:00-6:55 M W	6:00-7:20 M W	6:00-7:50 M W	6:00-6:55 T R	6:00-7:20 T R	6:00-7:50 T R	6:00-6:55 M W	6:00-7:20 M W	6:00-7:50 M W	6:00-6:55 T R	6:00-7:20 T R	6:00-7:50 T R	* 6:00 - 7:50 #6 Lab Block
	7:00-7:55 MW			7:00-7:55 T R			7:00-7:55 MW			7:00-7:55 T R			
#8 Class Block starts 8:00 p.m.	8:00-8:55 M W	8:00-9:20 M W	8:00-9:50 M W	8:00-8:55 T R	8:00-9:20 T R	8:00-9:50 T R	8:00-8:55 M W	8:00-9:20 M W	8:00-9:50 M W	8:00-8:55 TR	8:00-9:20 TR	8:00-9:50 T R	* 8:00 - 9:50 #7 Lab Block