

## CS3342 Course Outline B Term 2018-2019 Academic Year

### 1. Course Information:

- *Course Calendar*
  - 7 Jan 2019 (Monday): Classes Resume
  - 9 Jan 2019 (Wednesday): First Meeting of CS3342b
  - 15 Jan 2019 (Tuesday): Last day to add a half course
  - 24 Jan 2019 (Thursday) -- **Warmup task due** in course git repository by end of day on 24 Jan 2019.
  - 14 Feb 2019 (Thursday) -- **Ruby task due** in course git repository by end of day on 14 Feb 2019.
  - 18 Feb 2019 -- 22 Feb 2019 : SPRING READING WEEK -- hopefully it is spring then
  - 7 Mar 2019 (Thursday) : Last Day to Drop (see university calendar to double check date if this is relevant to you)
  - 7 Mar 2019 (still Thursday) -- **Prolog task due** in course git repository by end of day on 7 Mar 2019.
  - 21 Mar 2019 (Thursday) -- **Erlang task due** in course git repository by end of day on 21 Mar 2019.
  - 9 Apr 2019 (Tuesday): Last day of classes in Winter Term
  - 9 Apr 2019 (still Tuesday) -- **Haskell task due** in course git repository by end of day on 9 Apr 2019.
  - 11 Apr 2019 -- 30 Apr 2019: Final Exam Period
- *Registrar's TimeTable (as of 3 Jan 2019):*
  - Wednesday, 3:30 pm - 5:30 pm; NCB-113
  - Thursday, 10:30 am - 11:30 am; NCB-113
- *Academic Calendar:*
  - Computer Science 3342A/B - Organization of Programming Languages
  - Specification and analysis of programming languages; data types and structures; bindings and access structures; run-time behavior of programs; compilation vs. interpretation. Comparative presentation of at least three programming languages addressing the above concepts.
  - Antirequisite(s): SE 3352A/B.
  - Prerequisite(s): Computer Science 2211A/B.
  - Extra Information: 3 lecture hours, 0.5 course.
  - *sidenote:* CS2211A/B covers the C programming language. It in turn requires CS1027A/B or CS2101A/B. CS1027A/B covers Java. CS2101A doesn't specify the language it uses -- different versions of it have used Matlab and C. So C would appear to be the common background language for the course.
- *Senate regulation:* Unless you have either the requisites for this course or written special permission from your Dean to enroll in it, you may be removed from this course and it will be deleted from your record. This decision may not be appealed. You will receive no adjustment to your fees in the event that you are dropped from a course for failing to have the necessary prerequisites
- *note:* As a computer scientist, you should recognize the above as a case of patching flawed software (in this case the registrar's online enrollment software) with `blame the user' policies. Hopefully you will never do anything like that in your future undertakings.

2. *Instructor information:* Robert E. Webber, MC384, webber@csd.uwo.ca (include course number in subject line), office hours on course announcements page (I am usually available immediately after class).

### 3. Course Syllabus

- In addition to the catalog course description above, I would add that each programming language represents a philosophy of how to program. Practice work in the course will focus on four languages: Ruby (object-oriented programming and metaprogramming); Prolog (logic programming), Erlang (functional language with scalable concurrency focussed on reliable systems) and Haskell (strongly typed functional programming). The class textbooks cover other languages as well. In addition to considering a wide variety of programming languages, we will also be interested in how they are implemented.
- The course format will involve assigned readings from the textbooks and practice work to more fully explore some of the ideas in the readings. Both the readings and the practice work will be subject of classroom discussion. The final exam will be primarily based on the textbook readings.

#### 4. Course Materials

- The course has two textbooks, both of which should be available in the campus bookstore:
  - [Seven Languages in Seven Weeks: A Pragmatic Guide to Learning Programming Languages](#); by Bruce A. Tate; Pragmatic Bookshelf; 2010+
  - [Introduction to Compiler Design 2nd edition](#) by Torben Mogensen; Springer-Verlag, 2017.

The final exam questions will come primarily from the course textbooks. Some relevant material will be provided on the course web page, blog, and wiki -- but where such material is relevant to the final exam, it will be clearly designated as such in the list of coverage for the final exam.

- *Course web page*: The course web page can be found at: <http://www.csd.uwo.ca/~webber/CS3342/index.html>. On it is a link to this course outline and a link to the course wiki where marking related information is sometimes posted. Marking is done on the assumption that people check the course wiki on a regular basis (in particular, the day before anything is due). Once Atlassian accounts have been set up for everyone who added the class by the add date, the course announcements will be found on the course wiki. This is a subarea of the departmental wiki <https://wiki.csd.uwo.ca/>. This course does not use OWL. Programming assignments are handed in using the course git repository (see course wiki for more information).

#### 5. Methods of Evaluation

- *Reminder: the marks in this class are meant to represent your individual effort -- these are not group tasks. Such clarifications as are necessary should be shared with the entire class to keep a level playing field -- either as questions in class or posted in the comment section of course wiki pages.*
- *Final Exam*: There will be a 2-hour final exam whose time will be announced later by the registrar's office. It will be a closed book, closed note, no electronics comprehensive exam. A detailed list of coverage will be provided as the semester develops. Specifically a database of questions and answers will be made available along with a program that randomly samples and shuffles the questions. The final exam counts 24% of the course mark.

As usual the Dean's office has to approve makeup exams for the final. Exams that start at the same time as the official final will be created from the same random sample of questions as the official final. Makeup exams (or any exams that have to be scheduled at a time different from the official offering) will be sampled from the same database as the final, but with a different random key for sampling and for shuffling.

In the event that any of the tasks outlined below end up being cancelled (due to nuclear war or whatever), the value of the final exam will expand to cover the cancelled tasks.

- There are five 'programming' tasks in this course: a warm-up exercise, and a programming assignment in each of the four languages of interest (Ruby, Prolog, Erlang, and Haskell). The warm-up exercise counts 4% of the final mark, the remaining four programming assignments count 18% of the mark each. Since all the assignments involve the functioning of the departmental git repository, in cases of exceptional unavailability of said repository, due dates will be extended and the extension announced on the course wiki. In the event that the course wiki is also not available, announcements will be made on the course web page.

The warm-up exercise involves transferring data back and forth from your git repository. More details will be posted to the wiki once the git repositories are available. The due date for the warmup is based on the notion that it can be done with no more than 5 days notice. If you don't have a git repository account 5 days before the due date you need to bring that to the attention of the prof while there is still time to fix it and allow you to make the regularly scheduled due date.

The assignments are due at the posted due date -- no automatic late extensions. Since they count so much of the mark, any exceptions have to go through the Dean's office. Shortly after the due time I will download the contents of all the course git repositories and the due assignment will be marked based on what is in each student's git repository at that point in time. Note that git is a versioning system and so automatically handles updates. What this means is that you can hand in your work as frequently as it reaches a new milestone and only the final handin will be marked.

Indeed not making frequent uploads indicates a lack of knowledge of how to properly use computer versioning systems and is one of the reasons why forgetting to upload your final version is not an excuse -- instead there should be a recent upload that is close enough to the final that the penalty for forgetting the final would be minor. Note you should not use branching in the git repository or other fancy features as it may result in the marker not being able to determine what you are handing in.

The assignment you are supposed to do will be placed in your git repository. Usually sample test data will be provided, but the program is expected to meet the task description and not just pass on the provided test data (i.e., the marking process will include other test data). There will also often be style restrictions, i.e., aspects of the language you are not allowed to use that will also be relevant to the calculation of the assignment mark. There will also be execution time restrictions. For full marks, you must program a solution in the specified language that seems reasonable to the prof, passes various test data in the expected time period, and has no style problems. Furthermore the solution must be submitted on time and following the submission instructions in order not to be penalized for wasting the marker's time. In particular, be sure you have the right file name in the right directory and, where specified, you implement the classes, methods, and function that are the interface to your work with the specified names and type signatures (to enable unit testing where appropriate).

#### 6. Additional Statements

- o *Statement on Use of Electronic Devices:* The final exam will be closed book, closed notes, with no electronic devices allowed, with particular reference to any electronic devices that are capable of communication and/or storing information.
- o *Statement on Use of Personal Response Systems:* Clickers will not be used in this class as it is nearly impossible to use them and protect student privacy, as any Computer Scientist should know.
- o *Statement on Academic Offenses:* Scholastic offenses are taken seriously and students are directed to read the appropriate policy, specifically, the definition of what constitutes a Scholastic Offence, at the following Web site: [https://www.uwo.ca/univsec/pdf/academic\\_policies/appeals/scholastic\\_discipline\\_undergrad.pdf](https://www.uwo.ca/univsec/pdf/academic_policies/appeals/scholastic_discipline_undergrad.pdf)
- o *Statement on Academic Offenses (graduate version):* Scholastic offenses are taken seriously and students are directed to read the appropriate policy, specifically, the definition of what constitutes a Scholastic Offence, at the following Web site: [http://www.uwo.ca/univsec/pdf/academic\\_policies/appeals/scholastic\\_discipline\\_grad.pdf](http://www.uwo.ca/univsec/pdf/academic_policies/appeals/scholastic_discipline_grad.pdf)
- o *Regarding Plagiarism-Checking Software:* Web-based versions of such software will not be used in this class as it is nearly impossible to use them and protect student privacy. In the case of both material handed in and tests and/or exams, any plagiarism checking will be done the old fashioned way, by a person looking at the material in question (with the aid of various special-purpose program analyzers).
- o *Senate regulation:* Students who are in emotional/ mental distress should refer to Mental Health @ Western <http://www.uwo.ca/uwocom/mentalhealth/>
- o *Support Services:* Please contact the course instructor if you require lecture or printed material in an alternate format or if any other arrangements can make this course more accessible to you. You may also wish to contact Student Accessibility Services (SAS) at 661-2147 if you have any questions regarding accommodations.

The policy on Accommodation for Students with Disabilities can be found here:

[http://www.uwo.ca/univsec/pdf/academic\\_policies/appeals/accommodation\\_disabilities.pdf](http://www.uwo.ca/univsec/pdf/academic_policies/appeals/accommodation_disabilities.pdf). There it tells you that such requests should first go to Services for Students with Disabilities (SSD) and that they make the initial determination of what is appropriate to then be followed up with the faculty member by them. It is also important for the student to contact the faculty member after having sorting things out with SSD to further sort out the operational details of what will be done.

Note: I also teach CS9622, a graduate course on nonfunctional software requirements with particular focus on accessibility (along with safety (security) and sustainability). So, if such matters are of interest to you, feel free to discuss them informally with the prof.

- o *Senate regulation:* For UWO Policy on Accommodation for Medical Illness and a downloadable SMC see: [http://www.uwo.ca/univsec/pdf/academic\\_policies/appeals/accommodation\\_medical.pdf](http://www.uwo.ca/univsec/pdf/academic_policies/appeals/accommodation_medical.pdf) Students seeking academic accommodation on medical grounds for any missed tests, exams, participation components and/or assignments worth 10% or more of their final grade must apply to the Academic Counselling office of their home Faculty and provide documentation. Academic accommodation cannot be granted by the instructor or department.
- o *link to medical form* [http://www.uwo.ca/univsec/pdf/academic\\_policies/appeals/medicalform.pdf](http://www.uwo.ca/univsec/pdf/academic_policies/appeals/medicalform.pdf)
- o *link to the website for Registrarial Services:* <http://www.registrar.uwo.ca>
- o *Link to services provided by the University Students' Council:* <http://westernusc.ca/services/>
- o *Department specific information*
  - [CS Department Rules of Ethical Conduct](#)
  - [CS Department Specific Information on Scholastic Offences](#)

## 7. Specification for this document

- o [course outline requirements](#)
- o [requirements regarding undergrad evaluation](#)
- o [requirements regarding scheduling of tasks](#)