Instructor:	Tamon Stephen
Meeting Time:	MWF 9:30–10:20 (remote)
E-mail:	tamon at sfu ca
Web page:	`http://www.sfu.ca/~tstephen/Teaching/1207_Math408/`
Office Hours:	Monday 10:30–11:20 (tentative) and by appointment.
Text:	Integer Programming by Conforti, Cornuéjols and Zambelli.
408 Grading:	50% Homework, 10% Midterm, 20% Computational Project, 20% Final
708 Grading:	40% Homework, 10% Midterm, 15% Computational Project, 15% Topical Project, 20% Final

1. **Remote Course Information.** Due to the Covid-19 pandemic, this is a **remote** course, that will be conducted entirely on-line. The tentative plan is to run the course through SFU's `Canvas` course management system. Students registered in this course should be able to log in to that at `http://canvas.sfu.ca` and find this course on their dashboard.

   In particular, lecture notes, assignments and perhaps videos will be posted on the course Website, along with details for connecting to class meetings. Assignments may also be submitted through `Canvas`.

2. **Class Meetings.** Class meetings will take place on-line MWF 9:30-10:20. The tentative plan is to do them via `Zoom`, a link will be provided in `Canvas`. These will not be lectures in the usual sense, but rather time for us to discuss the course material and for students to ask questions. These meetings will **not be recorded**. Please do not record them yourselves out of consideration for your fellow students who do not want to be recorded. However, please do attend the meetings!

3. **Syllabus.** This course is an introduction to discrete optimization. The focus is on modelling problems as integer programs and polyhedral methods for solving these programs. Topics that we plan to cover include:

   Model building using integer, binary and mixed integer variables. Computer solution of integer programming models, linear programming relaxations, Lagrangian relations, duality, simple upper bounds using greedy algorithms. Branch and bound algorithms, implicit enumeration, LP based branch and bound.

   Valid inequalities, Gomory's fractional cut, mixed integer cuts, strong valid inequalities, simple facets for 0-1 knapsack polytope and the travelling salesman polytope, branch and cut algorithms.

   Lagrangian relaxation, Lagrangian dual, Lagrangian heuristics. Column generation algorithm, solving symmetric travelling salesman problem using column generation.

   Greedy and local search algorithms, construction heuristics, worst case analysis of heuristics.

4. **Individual Homework.** There will be five individual homework assignments during the term. Late homework will not be accepted.

   You are encouraged to talk with each other and the instructor about the homework, but you must write up the solutions yourself, using your own words. Solutions copied from other students, textbooks or the Internet are **not** acceptable.

   Note that model solutions to homework problems will **not** be provided, even after the fact.

   Assignments, projects and exams in this class require well-written solutions.

5. **Computational Team Projects.** In addition to the individual Homework, students will work on larger-scale computational questions in teams. Teams will be assigned in the second week of classes, and the projects will proceed in stages, in roughly biweekly increments. Details will be posted to Canvas.

6. **Graduate student projects.** Near the end of the term, each graduate student will present a brief (25 to 30 minute) introductory lecture on a current topic in integer programming. The topic will be selected in conjunction with the instructor. Possible sources of topics are the book *Algebraic and Geometric Ideas in the Theory of Discrete Optimization* by De Loera, Hemmecke and Köppe or the surveys from *50 Years of Integer Programming 1958-2008* by Jünger. There may be an option to give these presentations in the SFU Operations Research Seminar series rather than in class.

7. **408 or 708?** Undergraduates who have already done well in 400-level Math courses and are considering graduate studies may in some cases consider taking the 700-level version of this course. If this applies to you, please get in touch with me after the first class.

8. **Computing.** Integer programming is a computational subject, and students are encouraged to experiment with software for integer programming. Some integer programming capability is now available even in general purpose software such as `Microsoft Excel`. There are also many specialized free and commercial packages for mathematical optimization.

   In this class, we will solve larger integer programs using the `AMPL` modelling language and environment, with popular commercial solvers such as `Cplex` and `Gurobi`. These are provided on an educational licence, and are only available for work on this course during the term. Instructions for downloading `AMPL` will be provided in class.

   The `AMPL` language comes with an on-line manual that is itself a good introduction to modelling with mathematical programs. It features examples and exercises that we will use in class, see: `https://ampl.com/resources/the-ampl-book/chapter-downloads/`.

9. **Exams.** Students **must** plan to take the tests at their scheduled times. The midterm will tentatively take place on `Monday, October 26th`. The final exam will take place during the final exam period (December 9th to 20th). SFU will announce the exact exam timeslots in mid-October. Note this is a departure from SFU's traditional practice of scheduling exams prior to registration.

   The tentative plan for the exams is to run them live over `Zoom`, following SFU's guidance for invigilation. Should the guidance be updated or the software prove inadequate, we may explore other alternatives for proctoring.

10. **Books.** The main textbook for this course is *Integer Programming* by Michele Conforti, Gérard Cornuéjols and Giacomo Zambelli. This book is available at the bookstore, and also on-line through the SFU library at: `https://link-springer-com.proxy.lib.sfu.ca/book/10.1007%2F978-3-319-11008-0`. (Requires login.)

    For background on linear programming, you may wish to consult Vanderbei's *Linear Programming: Foundations and Extensions*. A nice overview of the development of integer programming is contained in the book *50 Years of Integer Programming 1958-2008*, edited by Jünger et al. *The Traveling Salesman Problem: A Computational Study* by Applegate, Bixby, Chvátal and Cook gives a view of modern computation on this challenging problem, developing from scratch, through techniques presented in this course, right to the cutting edge (of ten years ago). It develops many of the techniques used in this course. All three of these books are available in electronic form through the SFU library.