McGill School of Computer Science
www.cs.mcgill.ca

# Course Outline

**Course Name**: Introduction to Computer Systems
COMP-273 Fall 2019

**Instructor**: Joseph Vybihal

**Contact Information**:

| | |
|---|---|
| Office: | ENGMC 323 |
| Office Hours: | TW 10:00-11:00 or by appointment |
| Email: | jvybihal@cs.mcgill.ca |
| Course email: | cs273@cs.mcgill.ca |
| Discussions: | https://piazza.com/mcgill.ca/fall2019/comp273/home |
| Website: | MyCourses |

**Course Objectives**: This course covers two major topics in computer systems: assembly language programming and the architecture of a CPU. Programmers are normally familiar with the computer only from a very high level, manipulating the computer using an object-oriented programming language. They are not aware of the many abstraction layers that hide the underlying structure and functions carried out by a computer, like the operating system, the machine language, and the underlying physical mechanics. This course peels back these layers of abstraction allowing the student to directly manipulate and study the engineering of some of the most important components of a computer. Our method of study will be two pronged: First, we will look at the engineering of the computer's CPU, RAM and various topics about the system-board. Last, we will learn the MIPS assembly language, as a tool to interact with the internal components of the computer. We will use MARS as our MIPS emulator and LOGISIM as our circuit emulator.

**Primary learning outcomes**: to get a clear understanding of how the CPU and RAM are built and function from the microcircuit and functional levels. To use an assembly language to interact directly with the hardware of a computer.

**Secondary learning outcome**: to demystify the abstraction layers in a computer system and to take advantage of the new-found capabilities. The student will be able to intelligently discuss the internal workings of computers and produce solutions that take advantage of the built-in hardware of a machine.

**Course Description**: Number representations, combinatorial and sequential digital circuits, MIPS instructions and architecture data-paths and control, caches, virtual memory, interrupts and exceptions, pipelining.

**Texts**:

**Primary Text:**

COMPUTER ORGANIZATION & DESIGN: The Hardware / Software Interface
Step 1: Create or sign into an account at http://learn.zybooks.com
Step 2: Enter zyBook code: **MCGILLCOMP273VybihalFall2019**
Step 3: Subscribe
Note: A subscription is $54 USD. Subscriptions last until January 3, 2020.

Texts are available on two-hour reserve in the Schulich Library. Call Numbers can be found from the McGill library website (see MUSE, Course Reserves).

**Optional Texts:**

(1) "See MIPS run", by D. Sweetman
(2) The Soul of a New Machine, Avon Books, ISBN: 0-380-59931-7

**Evaluation**:

| | | |
|---|---|---|
| Assignments | 30% | 6 Assignments (5% each) |
| Class Tests 1 | 10% | TBD |
| Class Test 2 | 10% | TBD |
| Final Exam | 50% or 100% | During regular final exam period |

You may use your final exam to replace you class test grades (100%).
The final exam will cover the entire course.

Grading: All software solutions must compile with zero errors and must run to be graded. It does not need to run correctly for grading, but it must run. If your program compiles with errors or does not run at all then you will receive zero points. The grader will not fix your code or look at the source code to give you partial grades.

Late work: You will be notified in advance of assignment due dates. All assignments are due on My Courses at the indicated time and date. Late assignments will lose 5% of its grade per day late. Assignments beyond 2 days late will not be accepted. You may not submit assignments via e-mail without the permission from the instructor.

Additional Work: Students with grades of D, F or J will not be given the opportunity to complete additional work to upgrade their grade.

Supplemental Exam: Check with the university for this opportunity.

Re-grading: Mistakes can occur when grading. Not surprisingly, requests for re-grading always involve those mistakes in which the student received fewer points than they deserved, rather than more points than they deserved. With that in mind: if you wish me to re-grade a question on an exam or assignment, I will do so. I reserve the right to re-grade other questions as well.

Cheating/Collaboration: Collaboration is encouraged but your discussions should be public in the sense that anyone including the professor should be allowed to listen in. Assignments are original works created by the student alone. You are permitted and encouraged to have conversations with other students concerning the contents of the assignments and how to do them, but your work must be original. If two or more assignments are found to be identical (or portions of assignments) then all parties will lose points. This includes the student who permitted their assignment to be copied. This includes written solutions and software source code. The students will be reported to the university plagiarism department.

**Co-Requisites**

* COMP 250 Introduction to Computer Science (unofficial, but strongly recommended)
* COMP 206 Introduction to Software Systems (official)

If you have not taken 206 or you are not taking it currently, then you should not take COMP 273, unless you have previous experience with C or C++ programming.

**Communication**

My Courses**:** All official communication, including announcements, lecture material, assignments, grades will be found on My Courses.

Course Discussions: The online free tool, piazza.com, is used as our course discussion board. Please make sure to enroll in the Fall 2019 COMP 273 course on Piazza. Use this as your primary communication medium, since your questions are public and can help other students.

Course Email: This is best for direct communication. Both the TA and professor use this account. Expect a reply in 24 hours or less. Use this email account for questions you do not want other students to see.

Private Email: The professor and TA have private email accounts that you may also use, however these communication channels are for personal queries. For example: if you have a problem with your grade then email the TA who graded you directly, do not email the prof and do not use the course email address.

Appointments: Please email me directly to book at appointment outside office hours.

Office Hours: I have posted office hours. Come to those times without appointment.

After lecture: Some optional time will be available just after class to ask questions. I do not guarantee the length of this time since other constraints may interfere.

```
CommunicationAlgorithm() :
    if (public) piazza(); // all will benefit
    else if (about marks) emailTAPrivately();
    else if (medical or special) emailProfPrivately();
    else courseEmailAccount();
```

**Your TA**

Each student is assigned a single TA who will be “their” TA for the entire course. This TA will be responsible for grading your assignments and this TA will hold weekly tutorials for their assigned group, which you can attend optionally.

Regardless, you can attend any TA office hour for help.

The table below identifies which TA you have been assigned to. The full details about all the TA s of this course can be found in myCourses.

Your TA will contact you to determine the best time to hold the weekly tutorial. It might happen that a common time is impossible to find. In that case, the TA will find a tutorial time that maximizes the number of attending students. Do not worry if you cannot attend. These tutorials are optional, and the work is posted on myCourses, so you can do them on your own and then go to any TA office hour for help.

**Tentative TA Group by Student Last Name**

| Last Name of Student | TA Name | TA Email Address |
|---|---|---|
| **A – JAC ………………. (85)** | Richard Olaniyan (2) | richard.olaniyan@mail.mcgill.ca |
| **JAM – LIS …………… (42)** | Jesse Islam | jesse.islam@mail.mcgill.ca |
| **LIU – RA ……………... (49)** | Wing Hang (Henry) Ho | wing.h.ho@mail.mcgill.ca |
| **RE – Wang A ………… (48)** | Ridwan Kurmally | ridwan.kurmally@mail.mcgill.ca |
| **Wang J – ZZ …………. (39)** | Tzu-Yang (Ben) Yu | tzu-yang.yu@mail.mcgill.ca |

254 Students and 6 TA units = 42.3 students per TA unit

# Tentative Course Schedule

| WEEK | TOPIC | READINGS | WORK HANDED OUT |
|---|---|---|---|
| **Unit 1 – BASICS** | | | |
| **1** | (A) INTRODUCTION<br>Introduction to the course and the instructor, undergrad research opportunities, about classical/pipeline CPUs and assembler programming, logisim & Mars, famous names | 1.1 to 1.3 | |
| | (B) THE SYSTEM BOARD<br>The computer, the system board and its parts, the CPU fetch cycle, bits/bytes (high-level view). Data flow through a computer. | 1.4 to 1.13 | |
| **2** | (A) NUMBER SYSTEMS & MEMORY FORMATS<br>Binary, hex, octal (in Bash), math operations, complement, conversions, basic data storage, basic data types, strings & arrays, IEEE floating point | 2.1 to 2.7 | Ass 1 – Numbers, Memory & Truth table circuits |
| **Unit 2 – CIRCUITS** | | | |
| | (B) INTRO TO CIRCUITS<br>Gates, wires, truth tables, truth tables to circuits, combinatorial circuits, ROM, Clock | 3.1 to 3.3 | Tutorial A: Truth table circuits |
| **3** | (A) RAM AND U-BUS INTRO<br>High and low level definition of RAM, the flip-flop, the D-flip-flop, bytes, addressing, example data storage, example 2-nibble RAM, Basic U-BUS (no addressing) | 3.6 to 3.8 | Ass 2 – RAM & U-Bus |
| | (B) REGISTERS & FULL U-BUS<br>High and low level definition, gate control (addressing and triggers), RAM/Register/Bus example, introduction to control unit triggers | 3.9 | Tutorial B: LOGISIM and truth table circuits |
| **4** | (A) ALU<br>High and low level definition, adder circuitry, subtraction circuitry, status register (logical operations), multiplication and division as a form of adding and subtracting. | 3.4 to 3.5 | Tutorial C: Registers, addressing and U-Bus circuits |
| | (B) CLASS TEST #1 | | CLASS TEST #1 weeks 1-4 |
| **Unit 3 – MICRO INSTRUCTIONS + CPU ELEMENTS** | | | |
| **5** | (A) WRITING MICRO INSTRUCTIONS<br>Macro instructions are made from multiple micro instruction steps, hand written syntax, examples with LW, SW, ADD, CMP, BEQ, J (using the system board) | Notes | Ass 3 – ALU & Micro instruction writing |
| | (B) BUILDING MICRO CIRCUITRY<br>Designing circuits that implement LW, SW, ADD, CMP, BEQ, J (using the system board) | 6.1 to 6.2 and 4.1 to 4.3 | Tutorial D: Micro instruction writing |
| **6** | (A) THE CONTROL UNIT + CPU ELEMENTS<br>High level definition, low level definition, controlling the micro instructions, examples connecting to previous lectures, the IR and instruction formats | 4.4 and 5.1 to 5.5 | |
| **Unit 4 – CPU TYPES & PERFORMANCE** | | | |
| **6** | (B) CLASSICAL CPU<br>High level definition, instruction execution flow, performance calculations | Notes | Tutorial E: Building control units |

| | | | |
|---|---|---|---|
| **7** | (A) PIPELINE CPU<br>High level definition, instruction execution flow, performance calculations, faults | 4.5 to 4.8, 4.13 | Ass 4 – CPU performance & The Control Unit |
| **Unit 5 – MIPS ASSEMBLER PROGRAMMING** | | | |
| **7** | (B) THE MIPS CPU<br>Registers, addressing modes, instruction syntax, writing a program, compiling a program, example hello world program, OS Interrupts introduction, co-processors | 11.2 to 11.3<br>9.1 to 9.5 | Tutorial F: CPU faults and performance |
| **8** | (A) CLASS TEST #2<br><br>(B) DATA & MATHEMATICS PROGRAMMING<br>Defining simple data, data accessing instructions, basic mathematical instructions, simple I/O, example programs | 6.3 to 6.4 | CLASS TEST #2 weeks 5-7<br><br>Tutorial G: Using MARS |
| **9** | (A) CONDITIONS & ITERATION<br>How to express a condition, a simple if-statement example, how to express a loop, a simple while-loop example, complex array conditions, example printing characters from a string<br><br>(B) SUBROUTINES & FUNCTIONS<br>Definition of a subroutine, definition of a function, examples, MIPS and ANSI parameter passing, the run-time stack, saving variables | 6.5 to 6.6<br><br>6.7 to 6.9 | Ass 5 – Basic MIPS programming<br><br>Tutorial H: MIPS programming |
| **10** | (A) More in class programming examples and problems | 6.12 to 6.13, 6.18 | |
| **Unit 6 – ADVANCED TOPICS** | | | |
| **10** | (B) CACHE<br>Basic architecture, modulo addressing, how it is used, performance, hit and miss, associative caches | 7.1 to 7.4 | Tutorial I: more MIPS programming |
| **11** | (A) EXTERNAL DEVICE I/O<br>Polling/Interrupts/DMI: High level definitions, performance, some example MIPS code<br><br>(B) VIRTUAL MEMORY<br>Virtual vs real memory, page tables, OS run-time loop | Notes<br>7.5 to 7.6<br><br>7.7 to 7.8 | Ass 6 – MIPS programming & performance<br><br>Tutorial J: Performance |
| **Unit 7 – If time permits** | | | |
| **12** | (A) MULTI-CORE CPU<br>Strategies, control flow, queues, the OS involvement, multi-threading, the programmer involvement<br><br>(B) FINAL EXAM REVEW<br>Talk about the final exam and what will be on it. Undergraduate research opportunities & about grad school | 10.4 to 10.5<br>4.10 to 4.12<br>Topics 8.1 to 8.5 | Tutorial for Final exam<br><br>Tutorial K: review for final exam |

# General Course Information

**Course Requirements**: COMP-206 is a co-requisite for this course.

If you already know how to program in C or C++ or received a very good grade in COMP 250 then you can probably take this course without the course requisite.

**Right to submit in English or French written work that is to be graded**

In accord with McGill University's Charter of Students' Rights, students in this course have the right to submit in English or in French any written work that is to be graded.

**Classroom Rules**: All electronic devices (cell phones and beepers) must be turned off or left on silent mode during class time.

**Assignments Pickup**: All assignments are submitted to and picked-up from My Courses.

**Computing Resources:** Trottier 3rd floor.

**Examination and Grading:**

Students are responsible for all materials for the tests and exams, whether it is covered in class. Exams will be a combination of all types of questions based on all sources, and students may be required to integrate theoretical concepts from the text to substantiate their arguments.

**No make-up tests or make-up assignments are allowed in this course**. A supplemental exam is possible for 50% of the grade (to replace your final exam).

If you are not satisfied with the grading of an assignment or mid-term test, you may request a review within 7 days of return. Indicate in writing or during a meeting with the instructor where and why you feel the marks are unjustified and give it back to your instructor for re-grading. Note that the entire assignment or mid-term test will be re-graded and your grade can go up or down (or stay the same) accordingly.

*Calculators*

Only non-programmable, no-tape, noiseless calculators are permitted. Calculators capable of storing text are not permitted in tests and examinations.

*Dictionaries*

Dictionaries are not permitted, but translation dictionaries are.

*Handheld Devices*

Handheld devices capable of storing text and having calculator functionality (e.g. Palm, etc.) are not permitted.

**Additional Information:** COMP-308 Computer Systems Lab in a continuation course to COMP-273. It is a hands-on assembler and C lab course that will teach students how to interface with actual computer hardware using low-level programming techniques. This may be of interest to you if you would like to continue learning how to interact at low levels.

The course slides are not meant as a complete set of notes or a substitute for a textbook, but simply constitute the focus of the lecture. Important gaps are left in the slides that are filled in during class, thus **l**ecture attendance should be considered essential.

The material covered in the classroom will be used to supplement textbook readings.

**Academic Integrity:**

*Code of Student Conduct*

McGill University values academic integrity. Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see *www.mcgill.ca/integrity* for more information).

L'université McGill attache une haute importance à l'honnêteté académique. Il incombe par conséquent à tous les étudiants de comprendre ce que l'on entend par tricherie, plagiat et autres infractions académiques, ainsi que les conséquences que peuvent avoir de telles actions, selon le Code de conduite de l'étudiant et des procédures disciplinaires (pour de plus amples reseignements, veuillez consulter le site *www.mcgill.ca/integrity*).

**Final Exam Policy**:

*Regulations*

Students should not make other commitments during the final exam period. Vacation plans do not constitute valid grounds for the deferral or the rescheduling of examinations. See the Centre Calendar for the regulations governing Examinations:
http://www.mcgill.ca/student-records/exams/regulations/
Students are required to present their I.D. Card (with photo) for entrance to their examination.

*Conflicts*

If you are unable to write your final examination due to scheduling conflicts, you must submit a Final Exam Conflict Form with supporting documentation at least **one month** before the start of the final examination period. Late submissions will not be accepted. For details, see
http://www.mcgill.ca/student-records/exams/conflicts/

*Exam Timetable*

Examination schedules are posted at the Centre and on the following page approximately 6-8 weeks before the examination period commences
http://www.mcgill.ca/student-records/exams/
The Centre cannot provide examination dates over the telephone.

**Email Policy**:

E-mail is one of the official means of communication between McGill University and its students. As with all official University communications, it is the student's responsibility to ensure that time-critical e-mail is accessed, read, and acted upon in a timely fashion. If a student chooses to forward University e-mail to another e-mail mailbox, it is that student's responsibility to ensure that the alternate account is viable.

Please note that to protect the privacy of the students, the University will only reply to the students on their McGill e-mail account.

**Students Rights and Responsibilities:**

Regulations and policies governing students at McGill University can be downloaded from the website:
http://www.mcgill.ca/deanofstudents/rights/

**Students Services and Resources:**

Various services and resources, such as email access, walksafe, library access, etc., are available to students:
http://www.mcgill.ca/stundet-records

**Minerva for Students:** http://www.mcgill.ca/minerva-students/

**Note**: In the event of extraordinary circumstances beyond the University’s control, the content and/or evaluation scheme in this course is subject to change.