Postgraduate New Student Welcome Afternoon

Eric Martin

School of Computer Science and Engineering
University of New South Wales
Sydney, Australia

24.02.2016
Congratulations for your admission to one of the CSE postgraduate coursework programs.

What we will do tonight:

- get to know each other and enjoy some drinks and snacks
- go over some important information
- examine on a case study how every course, starting with the foundation ones, can potentially provide you with useful skills and knowledge to develop innovative technology and a successful business
- examine how the courses fit together, by reflecting on the field and seeing the big picture
The exemption exams

- They have been held already, but you will have another opportunity to take them if needed. **They should be taken at the beginning of your studies**, not when you start your second semester.

- They apply to the following courses:
  - COMP9020  Foundations of Computer Science
  - COMP9021  Principles of Programming
  - COMP9024  Data Structures and Algorithms
  - COMP9032  Microprocessors and Interfacing
  - COMP9311  Database Systems
  - COMP9414  Artificial Intelligence
  - COMP9331  Computer Networks and Applications

- To apply to any exemption exam, you need to fill a **form** that you access from the CSE website.

- To get advanced standing or exemption for COMP9024 or COMP9331, you have to get advanced standing or exemption for COMP9021.
Important deadlines

- You have till the end of week 1 to enrol into a program.
- You have till the end of week 1 to make any change to your course selection.
- You have to pay your fees by the end of week 1; otherwise you will be disenroled.
- If you have any financial problem and cannot pay your fees then contact Commonwealth Support and Fees as soon as possible, by email (csandfees@unsw.edu.au) or phone (9385 3119).
- You have till the 31st of March to drop a course without financial penalty.

Attend the first lecture of as many courses as possible you are interested in and can enrol in, read the course outlines, so as to maximise the chances of making the right course selection.
How to select major and courses

Know yourself!

- Know what your interests are.
- Know your strengths and weaknesses.
- If you are passionate with your studies, and select major and courses based on your interests and your strengths, you will enjoy your studies
  and then
  get very good marks
  and then
  impress potential employers
  and then
  get a rewarding, well-paid job.
Constraints

- Be aware of program requirements:
  - You must take GSOE9820 Engineering Project Management, unless you have completed one management course as part of your undergrad studies, or unless you have extensive work experience.
  - You have to select at least 6 courses from the list *Advanced Disciplinary Knowledge Courses*, which expose you in one way or another to research.

- Know which courses are core in the majors of interest
- Be aware of prerequisites and corequisites
- Be aware that some courses are offered in either session 1 or session 2, or neither...
- Be aware of possible time clashes.

**Plan ahead, while still remaining flexible**
Options

- You can get advanced standing for completed courses that have not been credited towards a degree.
- You can get exemption for courses that are core in your program, but that you have done already.
- Be aware of the option of taking a UNSW free elective.
- Be aware of the option of substituting some CSE courses by some approved nonCSE courses, one (free or approved) nonCSE course for every 4 CSE courses.
- There is an option to take research projects in the last semester of study; this option is subject to a number of extra conditions.
Working on a course

- Every course has lectures. Some courses have labs.
- Every course requires **at least** 12 hours of study per week on average, with many courses requiring substantially more time.
- Being a full-time student is equivalent to having a **demanding full time job where you often have to work lots of extra hours**.
- You will mainly learn by tackling (often challenging) assignments, many of which require **tens of hours** of work.
- The understanding you will get from attending lectures will be shallow. Practical exercises and assignments will help you master the material, gain a deeper understanding.
- Assignments are essential to acquire technical and problem-solving skills.
- Work steadily, from the very first week. Start working on the assignments as soon as they are released.
- It is convenient to own a PC or a laptop and work at home, but you can book working stations in one of our labs.
Passing a course

- Every course has its own magic formula.
  - Some use arithmetic means, others harmonic means;
  - some require you to pass one or more assessment components; getting a total mark of 50 might then not be enough to pass.

- Read in detail the course outlines. They tell you everything on how the courses will be assessed.

- There is sometimes a mid-term exam.

- There is almost always a final exam, which can be written, or practical, or both. It is usually 2-3 hour long and might include multiple-choice questions, some discussion, some programming...

- If you have been working steadily during the whole session, exams should be pieces of cake.

- You should not aim at just “getting a pass.” Employers know that a grade of 50 or hardly more means...
Be informed, get more advice

Postgraduate Coursework Advising Session

Not sure of the services available to you at UNSW? Want to network with your peers? Want some advice from your School office?

Attend the Postgraduate Coursework Advising Session. Open for all new and current postgraduate coursework students from Civil and Environmental Engineering and Computer Science and Engineering.

**Date:** Monday 14th March 2016  
**Time:** 4:00pm-6:00pm  
**Location:** Engineering Design Studio- Ainsworth Building  

Canapes and Drinks Provided.

Register via the below link:
https://my.cse.unsw.edu.au/pg_advising.php
The problem: real time search from Twitter

- 289 million users do or not follow each other, have friends, possibly belong to lists, and converse or exchange information.
- A tweet contains at most 140 characters, with possibly embedded hashtags or urls.
- The author of a tweet can retweet someone else’s tweet (RT @uid is inserted at the beginning of the tweet), or reply to him or her (@uid is inserted at the beginning of the tweet), or mention him or her (@uid is inserted elsewhere in the tweet).
- The most valuable information is in the recent, not indexed yet, web documents linked to by the urls.
- The quality of the information is assessed by analysing signals associated with the source.

```
tweet-dumps-csv
```
Graph properties

- degree centrality of a node (the higher the number of ingoing and outgoing edges), the more central the node is
- nodes are *hubs* (many outgoing edges) or *authorities* (many ingoing edges)
- Global properties (e.g., *assortativity*, a measure of the likelihood for nodes to connect to other nodes with similar degrees)

**COMP9020  Foundations of Computer science**

Figure 15: Retweet trees of 'air france flight' tweets
A bit of research on social choice theory

Every user ranks other Twitter users from the ones he or she retweets most to the ones he or she retweets less.

Arrow’s theorem on the nonexistence of a perfect voting system:

*There is no voting system that satisfies these 5 conditions:*

- Every order is admissible.
- Changes favourable to a candidate on individual preference ballots do not cause that candidate to decrease in rank on the resulting societal preference order.
- The societal preference between any two candidates depends only on the individual voter’s preferences between those two.
- There is no pair of candidates one of which is preferred to or tied to the other regardless of how any of the voters vote.
- There is no voter $v$ such that for all pairs of candidates $A$ and $B$, if $v$ prefers $A$ over $B$ then society also prefers $A$ to $B$. 
Number of tweets

Number of users

Number of tweets

1.04858e+06

32768

1024

128

32

1

1
4
16
64
256
1024
4096
Number of followers

![Graph showing the relationship between number of users and number of followers. The graph displays a descending trend where the number of users decreases as the number of followers increases.]
Retweets versus mentions (1)

Number of times retweeted vs. Mentions

- Number of times retweeted: 0.5, 1, 2, 4, 8
- Mentions: 1, 32, 1024, 32768, 1.04858e+06, 3.35544e+07

CSE (UNSW)
Postgrad Welcome Evening
24.02.2016 17 / 38
Retweets versus mentions (3)
A look at the scripts

nb_of_tweets.pig
nb_of_followers.pig

COMP9021 Principles of programming

nb_of_times_retweeted.pig

COMP9041 Software construction: techniques and tools

nb_of_retweets_versus_nb_of_mentions.pig

COMP9311 Database Systems

Processing large data using the MapReduce model of distributed computation on a cluster of computers thanks to Hadoop software.

COMP9321 Web applications engineering
What is computer science and engineering?

It is useful to think of a field as having an **object of study** and possibly developing **applications** based on the **knowledge** acquired by studying this object.

The objects of study of cosmology, biology, optics, chemistry, and their applications, if any, are pretty clear.

What is the object of study of computer science and engineering?
When the field began

The study of **computation**, that is, essentially, the study of the so called **computable functions**.

Which of these functions are computable?

1. The function that given two integers as input, returns their sum.
2. The function that returns 1 if there will still be human beings on Earth in year 3,000, and returns 0 otherwise.
3. The function that given a C program that compiles and that receives no input, returns 1 if the program crashes, waits for input or outputs something after some time, and 0 if it loops.
4. The function that given an integer \( n \) as input, returns the \( n \)th digit after the decimal point in the decimal expansion of \( \pi \).
5. The function that given two real numbers as input, returns 1 if these numbers are equal and 0 otherwise.
A key result is the existence of a *universal* computable function, which allowed one to envision programmable devices to solve arbitrary (computable) problems, rather than devices dedicated to solving a unique (computable) problem.

**COMP4141 Theory of computation**

Some problems have a computable solution *in theory*, but not in any practical term. Complexity theory addresses these issues. Algorithmic theory proposes practical computable solutions to many problems in a rather systematic way, considering families of (generic) problems and families of (generic) solutions, possibly only approximated.
Application: the computer

The computer is one of our main tools, similarly to the fact that a telescope is one of the main tools of an astronomer. But whereas the latter is an application from optics, the former is an application of our very field (and others, such as electronics).

A computer has to be designed and built.

- COMP9032 Microprocessors and interfacing
- COMP9222 Digital circuits and systems
- COMP9211 Computer architecture

It has to implement a universal computable function.

- COMP9102 Programming languages and compilers
Making it easier to speak to a computer

It needs high level programming languages to implement the solutions to specific problems. These languages can be classified into families and studied at an abstract level.

**COMP9161 Concepts of Programming Languages**

In practice, we will use particular instances of the former and program in some specific language.

**COMP9021 Principles of programming**
**COMP9024 Data structures and algorithms**
**COMP6771 Advanced C++ Programming**
To organise all data and programs than can be run on a computer and execute programs efficiently, a special program, the operating system, needs to be designed and implemented.

COMP9201 Operating systems
COMP9242 Advanced Operating Systems

Operating systems must also be able to let many machines share the work

COMP9243 Distributed Systems
Managing the complexity of very large programs

Algorithmic theory is about coming up with very smart solutions to small, well defined problems. But many problems do not require so much clever solutions, but mainly a huge number of lines of code, with the assurance that the result meets the specifications, is robust, error-free, secure, etc. This leads to the field of **Software engineering**.

Dealing with writing good software that is easy to read and maintain, and using techniques that make it secure and reliable:

- **COMP6721**: (In-)formal methods: the lost art
- **COMP9181**: Language-based software safety
- **COMP9153**: Algorithmic Verification

Dealing with writing large programs:

- **COMP9041**: Software construction: techniques and tools
Making computers user friendly

COMP9511 Human Computer Interaction
Writing special kinds of programs

Programs that draw geometric or animated figures.

- COMP9415 Computer graphics
- COMP9018 Advanced graphics

Programs that play games

- COMP4431 Game Design Workshop

Programs that fulfil the needs of businesses (which requires of course understanding these needs, hence programming is only one aspect of the following courses; in particular, designing appropriate systems is crucial).

- COMP9321 Web applications engineering
- COMP9322 Service-oriented architectures
- COMP9323 e-Enterprise project
The computer and the brain

Many believe that the brain is a computer; others believe that the brain can be simulated by a computer (anything the former can do, the latter can do as well). So let us try and let computers do what we do. This leads to the field of **Artificial intelligence**.

We can try and come up with a (simplified) model of the brain and write programs that simulate its activity.

**COMP9444 Neural networks**

One can also completely ignore the workings of the brain and just focus on what we want the computer to achieve. Human beings are able to complete such a broad range of tasks.

**COMP9414 Artificial intelligence**
Focusing on particular tasks...

How can computers gain knowledge, how can we make them reason?

COMP4418 Knowledge Representation and Reasoning

How can computers prove theorems?

COMP4161 Advanced topics in software verification

How can computers build representations and analyse what they “see”?

COMP9517 Computer vision

How can they make a robot move and execute the right set of actions?

COMP4411 Experimental robotics

How can they make a robot plan its actions?

COMP9431 Robotic software architecture
...and on more particular tasks

How can they build a model of their environment, that evolves as the latter changes?

COMP9417 Machine Learning and Data Mining
COMP9418 Advanced Topics in Statistical Machine Learning
Are we still in the realm of *computable functions* and their applications?

More likely, our object of study and the applications we develop on the basis of the knowledge we gain on that object has something to do with **information**.

*From wikipedia:* Information is any kind of event that affects the state of a dynamical system. In its most restricted technical sense, it is an ordered sequence of symbols. As a concept, however, information has many meanings. Moreover, the concept of information is closely related to notions of constraint, communication, control, data, form, instruction, knowledge, meaning, mental stimulus, pattern, perception, and representation.

Information is an object of study in physics and communication theory, with **entropy** as a key notion. Some aspects of it are our object of study.
Communicating information (1)

How do computers communicate information to each other or to or from other devices, and in particular, using the Internet and using wireless technology?

- COMP9331 Computer networks and applications
- COMP9332 Network routing and switching
- COMP9333 Advanced computer networks
- COMP9334 System capacity planning
- COMP9335 Wireless mesh and sensor networks
- COMP9336 Mobile data networking
- COMP9337 Mobile data networking
- COMP6733 Internet of things experimental design studio
Communicating information (2)

How do we schedule the exchange of information?

COMP9151 Foundations of concurrency
COMP9152 Comparative concurrency semantics

How do we make sure that the exchange of information is secure?

COMP9441 Security engineering
COMP9447 Security engineering workshop
Storing and extracting information

How do we efficiently store, search for, retrieve, and perform fundamental operations on pieces of information, and in particular, pieces of information stored on the web?

COMP9311 Database systems
COMP9315 Database systems implementation

How can we compress the representation of a piece of information as much as possible and search from the compressed form?

COMP9319 Web data compression and search

How do we deal with and exploit massive amounts of data?

COMP9313 Big Data Management
Discovering information

How do we discover information from data?

COMP9318 Data warehousing and data mining

How do we make sure that the information we extract is relevant?

COMP6714 Information Retrieval and Web Search

Discovering information from biological data:

BINF9010 Bioinformatics Methods and Applications
BINF9020 Computational Bioinformatics
Conclusion

- CSE courses are demanding, but ultimately rewarding
- Work very hard, starting Monday!
- Devote some time to every course at least every second day, and preferably every day
- Never think that you will catch up later, seek help regularly whenever you need

Have a great time as a CSE student!