3rd Year Tips

**Do the CA** for all modules. It takes a huge amount of stress off the exams. None of them are too time-consuming on their own (unlike second year’s Computer Architecture).

Keep up to date in the modules. It is easy to fall behind in multiple ones and have a horrific time cramming before exams. It is **not** doable, and is unlike first and second year cramming.

# Semester 1

### CS3011 Symbolic Programming

Tim just teaches the content from [Learn Prolog Now](http://www.learnprolognow.org/lpnpage.php?pagetype=html&pageid=lpn-html), so if you can follow along and do the exercises from chapters 1 - 10 you will be fine.

### CS3016 Functional Programming

Again, like Symbolic Programming, the lecture content follows closely with [Learn You A Haskell](http://learnyouahaskell.com/chapters). If you know the contents of this book, you will do well.

### CS3021 Computer Architecture

Exam took place at Christmas for us, but was quite a theory-heavy exam.

Need to know the whole course in order to be safe.

### CS3041 Information Management

Like Computer Architecture, this exam took place at Christmas.

There is quite a lot of content in the course, but none of it is very difficult.

The group work portion of this module only exists for two weeks, before going back to individual work. It makes no sense.

### CS3071 Compiler Design (Difficult)

This course is actually quite short compared to other modules.

The difficulty in this course is getting your head around what is the final version of what needs to be learned. Dave Abe builds up to final concepts, and the prototypes you learn along the way will **not** be acceptable for the exam.

[OJ ‘s notes](http://dajuice.netsoc.ie/) are great for seeing exactly what you need for the exam.

**Assignment**:

If you finish the final part of the assignment, you will have an interview with Dave Abe in May. Here’s a few tips in order to do well:

1. **Comment** your code well (Not too much, not too little).
2. Don’t forget to keep the **HTML file** up to date.
3. Make a good **test program**:
   1. Remove the comment at the top of the file.
   2. Write to the console before and after each test, so that Dave can see that each test case is beginning and ending.
4. Make sure your **report** is detailed :
   1. Why did you make each decision you made?
   2. What branching decisions were made etc.

**Resources**:

[OJ ‘s Notes](http://dajuice.netsoc.ie/)

[YouTube Series](https://www.youtube.com/playlist?list=PLEbnTDJUr_IcPtUXFy2b1sGRPsLFMghhS) (Lectures 1 - 8)

# Semester 2

### CS3014 Concurrent Systems

The course might change next year, but you could get away with Q1 and Q2 in the exam just knowing OpenMP and Intel SSE intrinsics.

For a different lecturer, the exam could become a lot more theory heavy, as there is a lot of content covered in the module.

### CS3031 Telecommunications (Difficult)

There is a huge amount of content covered in this module, especially in the **cryptography** section. If you can remember the crypto algorithms, you can do very well in Q2, but otherwise it may be more worthwhile learning the other content well.

**Assignments**:

It is very easy to pick up 100% in the assignments if you simply have something resembling every key feature. Documentation seems to be worth nothing apart from plagiarism checking.

### CS3061 Artificial Intelligence (Difficult)

The only reason this module is in any way difficult is because Tim cannot lecture, and has awful lecture notes. You are better off learning topics from elsewhere, as the lecture slides do not follow logically, with no clear distinction on what topic you are currently reading about.

The assignments seem to factor into the exam very frequently.

Ask Tim for exam tips, and go to his revision lecture. He gives away way too much information, and will tell you what has a high chance / no chance of coming up.

**Recommendations**:

[Markov Decision Process YouTube Series](https://youtu.be/Jk2V9yA82YU)

[Q-Learning](http://mnemstudio.org/path-finding-q-learning-tutorial.htm)

### CS3081 Computational Mathematics (Difficult)

The reason this course is so difficult is because of its **length**. No one topic is particularly hard.

The lecture notes are entirely based on Numerical Methods for Engineers and Scientists (PDF is in this Drive).

Éamonn gives **no assignments** during the semester, so it is easy to end up learning the course for the first time just before exams. Don’t do that. I did that.

Do the exercises he recommends from the book at the end of each chapter. Éamonn bases his exam questions off of these, so if you can do these, you should be fine.

Be familiar with which formulae are in the **log tables** (PDF is in this Drive), since this can take the stress off learning some formulae such as:

1. Trapezoidal Rule
2. Simpson’s 1/3 Rule
3. Vector / Euclidean Norm
4. Inverse of a 2x2 Matrix
5. Taylor Series
6. Newton’s Method for solving a nonlinear equation

If Éamonn gives out a **sample paper**, there is a high chance that the real exam will be very similar.

**Learn Matlab**. It’s easy, but it will not be taught to you, and it will come up in the exam. Try solving some of the problems that are given to you in Matlab.

### ST3009 Statistical Analysis (Difficult)

The course is very manageable up until the final few chapters (chapter 16 onwards). It is important that you remain able to do the early questions late in the semester.

Doug gives a midterm test after week 6, which is worth 10%.

After this midterm, he gives **no more assignments**, so it is easy to fall behind here.

I don’t think that you are allowed to use log tables for the exam, but in case that changes, there are plenty of useful formulae in there that you would need to learn otherwise.