

Weekly Notes 2020

Bookmarks

[Thursday 15 October - Week 1 Term 4](#)

[Thursday 24 September - Week 10 Term 3](#)

[Thursday 17 September - Week 9 Term 3](#)

[Thursday 3 September - Week 7 Term 3](#)

[Thursday 27 August - Week 6 Term 3](#)

[Thursday 20 August - Week 5 Term 3](#)

[Factorising the product of two large prime numbers.https://www.youtube.com/watch?v=M7kEpw1tn50](https://www.youtube.com/watch?v=M7kEpw1tn50) This video is a quick introduction. I suggest that you watch it several times a piece of paper alongside you as you work out is going on. The individual math steps are easy(ish) but do build up.

[Thursday 13 August - Week 4 Term 3](#)

[I will post a discussion question in piazza about this weeks work. Please contribute to it.](#)

[Thursday 6 August - Week 3 Term 3](#)

[Thursday 30 July - Week 2 Term 3](#)

[Thursday 23 July - Week 1 Term 3](#)

[Thursday 2 July - Week 12 Term 2](#)

[Thursday 25 June - Week 11 Term 2](#)

[Thursday 18 June - Week 10 Term 2](#)

[Thursday 11 June - Week 9 Term 2](#)

[Appointments with me](#)

[Thursday 21 May - Week 6 Term 2](#)

[Appointments with me](#)

[Thursday 14 May - Week 5 Term 2](#)

[Appointments with me](#)

[Bits and Pieces:](#)

[Project Management](#)

[Managing Code](#)

[Thursday 7 May - Week 4 Term 2](#)

[Some more JavaFX tutorials:](#)

[And some Swing ones:](#)

[Thursday 30 April - Week 3 Term 2](#)

[GUI's](#)

[How to build GUI's that interact with the program backend](#)

[Java Swing vs JavaFX](#)

To Do

[Thursday 23 April - Week 2 Term 2](#)

[Thursday 16 April - Week 1 Term 2](#)

[Thursday 26 March - Week 8 Term 1](#)

[Thursday 19 March - Week 7 Term 1](#)

[Thursday 12 March - Week 6 Term 1](#)

[Thursday 5 March - Week 5 Term 1](#)

[Reminder - any problems doing the problems, use the group forum, or ask for an extra tutorial. I will contact your eDean after tomorrow's tutorial if you are behind and have not contacted me separately to get some extra help to catch up! 23 and 28 should have been completed - ignore the first two rows they are my accounts.](#)

[Thursday 27 Feb - Week 4 Term 1](#)

[\(Actual lesson - 9am Friday 21\) Reminder - any problems doing the problems, use the group forum, or ask for an extra tutorial](#)

[Thursday 20 Feb - Week 3 Term 1](#)

[\(Actual lesson - 9am Friday 21\) Reminder - any problems doing the problems, use the group forum, or ask for an extra tutorial](#)

[Thursday 13 Feb - Week 2 Term 1](#)

[Reminder - any problems doing the problems, use the group forum, or ask for an extra tutorial](#)

[Before Thursday 13 Feb - Week 1 Term 1](#)

Scroll down - most recent notes are at the top

Bookmarks

The meetings can be accessed at

Thursday 2pm Zoom <https://zoom.us/j/640927564>

Friday 11am tutorial (optional) same as above

For 1:1 meetings we will use this google hangout

<https://meet.google.com/psa-dkuu-csa>

using Chrome only as your browser.

Forum netnzcsp2020@googlegroups.com

Course site <https://sites.google.com/netnz.school.nz/csp>

Direct link to repl revision <https://repl.it/student/classrooms/118160>

Java cheatsheet <https://introcs.cs.princeton.edu/java/11cheatsheet/>

[3.7 and 3.8 assessment](#) (project management/programming)

[3.2 assessment](#) (user experience)

[UX checklist](#) (can use if you wish but the assessment overrules everything in here)

[Programming checklist](#)

Month	Events
July	20/7 Start of Term 3
August	7/8 Assessments due in 13/8 Start work on externals (we have 6 or 7 weeks to complete this)
September	25/9 End of Term 3 Ideally internals finished
October	12/10 Start of Term 4 23/10 Final date for external CAT (Date to do this is likely to be week of 19-23 Oct - I need to finalise this soon)
November	13/11 Absolute deadline for any outstanding internals

EXTERNAL Thursday 22 October or Friday 23 October

Name	Day, Date & Time
Matthew Lang (Timaru)	Friday October 16 time?
Mikael (Pompalier)	Friday 23 October time?
Marshall (Ashburton)	Friday 23 October time?
Arlo	Thursday 22 October time?
Andrew	Thursday 22 October time?
Dakin	Thursday 22 October time?

Thursday 15 October - Week 1 Term 4

Last full lesson today (as some of you are doing the externals tomorrow/next Thursday or Friday)!

Computer Science External - Complexity & Tractability - example answers here:

<https://docs.google.com/document/d/1H8JBbL7wa-jRmg-VnoLip2seRnyXxdgUjgESFsoMOyQ/edit#heading=h.7wskmxuuetkz>

Though the stuff from 17 September covers everything as well.

Reflection on your project external (91909: Present a reflective analysis of developing a digital outcome)

Two area you need to have thought about beforehand are:
two of the following implications:

- legal
- sustainability and
- health and safety.

2019 past paper <https://www.nzqa.govt.nz/nqfdocs/ncea-resource/exams/2019/91909-exm-2019.pdf>

Thursday 24 September - Week 10 Term 3

It's been a long term. Some of you have completed the internals, while some of you are still busily working at them. I am available to mentor you in these final weeks to ensure that all of you get the grades you deserve. Please contact me by email for this help - including over the next 2 weeks.

Today we will complete our answers for the Computer Science external. And may be even get on to the project external.

Thursday 17 September - Week 9 Term 3

Catch Ups for you all - next Thursday 24 September

<https://calendar.google.com/calendar/u/0/selfsched?sstoken=UUpQWXVKcW1fWGhkfGRIZmF1bHR8MTg1YTMzNGM0NmM4NDQzN2JhZjNIZDczOTgzMjkxY2Y>

Thursday 3 September - Week 7 Term 3

Catchups re internals

If you need an appointment with me this week, please let me know by email.

External - lets re go over quantum computing:

Quantum computing https://www.youtube.com/watch?v=4q0YG_5GXm0

- Explain in simple language what quantum computing is
- Explain how it might help with intractable problems

There is a whole branch of computer science dedicated to tractable/intractable problems. (Note the P, NP work is high Merit/Excellence)

This short video introduces the concept <https://www.youtube.com/watch?v=dJUEkxyIBw>
The heart of this is does $P = NP$ or $P \neq NP$

One more video - <https://www.youtube.com/watch?v=9MvbNPQiEE8>

I think it is time that we applied some of our knowledge about complexity and tractability in a way that might be useful for the external!

Here is the link to the 2019 paper - (this year's paper is likely to be similar)

<https://www.nzqa.govt.nz/nqfdocs/ncea-resource/exams/2019/91908-exm-2019.pdf>

Here are the questions:

- (a) Explain the key aspects of your chosen computer science area.
- (b) Explain the relevant algorithms or mechanisms that support your chosen computer science area.
- (c) Explain how your chosen computer science area is used, implemented or occurs. Use examples to support your answer.
- (d) Explain the key problems or issues related to your selected computer science area, AND how these have been, or maybe, addressed.
- (e) Provide a detailed explanation of how the technical capabilities and limitations of your chosen computer science area relate to humans. Use examples to support your answer.
- (f) Compare and contrast different perspectives on your chosen computer science area.
- (g) What conclusions can you draw about your chosen computer science area? In your answer, you could:
 - (i) explore less-obvious implications
 - (ii) justify predictions that you make
 - (iii) consider potential improvements
 - (iv) suggest innovative and imaginative wider uses.

Note that the order of questions roughly goes Achieved, Merit, Excellence.

Task - Using the [External draft bullet points as a guide](#), write a complete answer for this external ie questions 1-6.

This uses an exemplar paper. How do the questions from last year match up with the exemplar paper?

Your answers should

- go in to more detail than the bullet points
- Be in complete sentences

Share this with me.

Thursday 27 August - Week 6 Term 3

Catchups re internals

If you need an appointment with me this week, please let me know by email.

For the external we will now look at a more complex problem - how complexity and tractability are used in modern encryption.

Modern encryption relies on intractability for it to be secure.

Throughout these explanations we are looking at the role of intractability. **Understanding encryption can get difficult very easily** - we are just looking at **explaining the role of intractability in encryption.**

What is encryption?

<https://www.csfieldguide.org.nz/en/chapters/coding-encryption/>

First of all a simple video about key exchange

<https://www.youtube.com/watch?v=U62S8SchxX4&feature=youtu.be>

In the following video, there is an explanation of a one way function (that is intractable). For the external, you do not need to explain the whole of this (it is difficult to understand - degree level), but seeing how this is done will help you explain in simple terms what a one way function is and how it works (ie given the sum of two large prime numbers it is intractable to calculate (factor) the original two primes. This is the basis of how we communicate securely over the internet (banking email etc)

https://www.youtube.com/watch?v=YEBfamv-_do&feature=youtu.be

The third video follows on from the first - again there is lots of (very) hard maths in here that we don't need to understand in detail - the key bit is about the intractability of prime factorization.

https://www.youtube.com/watch?v=wXB-V_Keiu8

=====

The final 'reference' <https://www.onebigfluke.com/2013/11/public-key-crypto-math-explained.html>

Shows a 'simple' explanation of the maths - the key steps are steps 1 and 2 and the part about **Why is this safe?** At the very end:

1. Pick two prime numbers:

$$p = 7$$

$$q = 13$$

2. Multiply them together:

$$n = p * q$$

$$n = 7 * 13$$

$$n = 91$$

....Lots of hard maths to make a one way function for exchanging keys for encryption

Why is this safe?

In this simple example it's totally not. You can trivially take $n = 91$ and guess p and q with simple [factorization](#):

```
>>> set((91 / i) for i in xrange(1, 91) if 91 % i == 0)
set([91, 13, 7])
```

Redo all the math above and you have the private key. *Doh~*

Now imagine if n was such a large number that it took a very long time to guess p and q :

```
>>> set((12031294782491844L / i) for i in xrange(1, 12031294782491844L) if 12031294782491844L % i
== 0)
```

This takes a really long time. Even fancy solutions on the fastest computer on Earth would take until the end of the universe. That's why transmitting secrets with public key cryptography is safe. That's also why [great leaps in prime number theory](#) are frightening / exciting.

Thursday 20 August - Week 5 Term 3

Catchups re internals

Please make an appointment with me for either later this afternoon, tomorrow or Monday. [Please book a slot in my calendar](#).

External:

Last week we looked at the travelling salesman problem and how the complexity of the problem scales exponentially so that over time it becomes intractable.

This week we will look at some other problems that are similar to this, that also become intractable quickly.

[The bin packing problem](#). The [interactive is here](#). Try out the two different heuristic methods.
[The knapsack problem](#). Try the different problems.

Factorising the product of two large prime numbers. <https://www.youtube.com/watch?v=M7kEpw1tn50> This video is a quick introduction. I suggest that you watch it several times a piece of paper alongside you as you work out is going on. The individual math steps are easy(ish) but do build up.

Thursday 13 August - Week 4 Term 3

External [91908 Analyse an area of computer science](#) 3 credits

The area we are looking at is a really interesting topic called complexity and tractability. I find it fascinating and I hope you do too.

Most of the work for this topic can be found in Chapter 11 of the CS field guide.

<https://www.csfieldguide.org.nz/en/chapters/complexity-and-tractability/> This week you will need to read the following:

[Read through and complete 11.1 of the CSFG](#) What's the big picture?

[Read through and complete 11.2 of the CSFG](#) Algorithms, problems, and speed limits

[Read through and complete 11.3 of the CSFG](#) Tractability

[Read through and complete 11.4 of the CSFG](#) The travelling Salesman problem

For each of the chapters have a really good read and do any of the interactive activities. At the end of this you should have an idea why some problems can't be solved **precisely** using an algorithm - ie we might have a good solution (determined using a **heuristic** algorithm) but we cannot be sure it is the correct solution (ie there might be a slightly better solution) - all because some problems scale very rapidly.

I will post a discussion question in piazza about this weeks work. Please contribute to it.

Thursday 6 August - Week 3 Term 3

External Entries:

You will sit the external on **Thursday 22 October or Friday 23 October unless your school has decided on another date.**

Please complete the table below that shows the date you will do the external

Name	Day, Date & Time
Matthew Lang (Timaru)	Friday October 16 time?
Mikael (Pompalier)	Friday 23 October time?
Marshall (Ashburton)	Friday 23 October time?
Arlo	Friday 23 October time?
Andrew	Thursday 22 October time?
Dakin	Thursday 22 October time?

Catchup Meetings with me: Next Thursday 13 August (optional your choice)

<https://calendar.google.com/calendar/selfsched?sstoken=UUpQWXVKcW1fWGhkfGRIZmF1bHR8MTg1YTMzNGM0NmM4NDQzN2JhZjNlZDczOTgzMjkxY2Y> Thursday 30 July - Week 2 Term 3

Thursday 30 July - Week 2 Term 3

External Entries:

You will sit the external on **Thursday 22 October or Friday 23 October unless your school has decided on another date**. Today I sent an email to your eDeans and you about external entries. We will check your entries in today's lesson and go through the things you need to know for the external.

Catchup Meetings with me: Next Thursday 6 August

<https://calendar.google.com/calendar/selfsched?sstoken=UUpQWXVKcW1fWGhkfGRIZmF1bHR8MTg1YTMzNGM0NmM4NDQzN2JhZjNlZDczOTgzMjkxY2Y>

Write ups

And now some of the additional parts of the internal project.

Relevant implications are in both the UX and Programming Project. The bottom of [this document](#) outlines what they are.

The [programming project checklist](#) outlines in **red** some of the additional things you need to do in your write up. These are **essential**.

Thursday 23 July - Week 1 Term 3

Welcome back! Term 3 is always a busy term - so here is the plan for the rest of the year.

Month	Events
July	20/7 Start of Term 3
August	14/8 Assessments due in 13/8 Start work on externals (we have 6 or 7 weeks to complete this)
September	25/9 End of Term 3 Ideally internals finished
October	12/10 Start of Term 4 23/10 Final date for external CAT (Date to do this is likely to be week of 19-23 Oct - I need to finalise this soon)
November	13/11 Absolute deadline for any outstanding internals

Term 3 10 weeks

- I need to catch up with each of you about your internals. [Please book a slot in my calendar](#) for either later this afternoon or tomorrow afternoon.
- I will outline the external requirements and what this actually is about.

Thursday 2 July - Week 12 Term 2

Last week I said:

Some of you need a tutorial on how to make either IntelliJ/Netbeans work with Github (remember Github is used for managing versions of your code). I posted this out on piazza earlier this week.

<https://calendar.google.com/calendar/selfsched?sstoken=UUpQWXVKcW1fWGhkfGRIZmF1bHR8MTg1YTMzNGM0NmM4NDQzN2JhZjNlZDczOTgzMjkxY2Y>

I need to ensure that each of you has a clear idea of what to do over the next few weeks. Please book an appointment with me for either today or tomorrow.

Thursday 25 June - Week 11 Term 2

Last week I said:

Some of you need a tutorial on how to make either IntelliJ/Netbeans work with Github (remember Github is used for managing versions of your code). I am going to make video tutorials of this tomorrow and will update you with the URLs of the videos.

I few things cropped up and I have still to do these. I aim to get it done asap.

Tomorrow many of you have appointments with me as part of mentoring of your projects. Please attend them on time. I am not chasing you up if you miss the appointment - it is your loss!

<https://calendar.google.com/calendar/selfsched?sstoken=UUpQWXVKcW1fWGhkfGRIZmF1bHR8MTg1YTMzNGM0NmM4NDQzN2JhZjNIZDczOTgzMjkxY2Y>

Thursday 18 June - Week 10 Term 2

Some of you need a tutorial on how to make either IntelliJ/Netbeans work with Github (remember Github is used for managing versions of your code). I am going to make video tutorials of this tomorrow and will update you with the URLs of the videos.

I need you to share your github repository with **tstornetnz** please.

Reminder - your project log - this should have screenshots and notes of your progress. Include improvements suggested by your stakeholder.

You will need to keep your Trello or Github Project up-to-date.

I am going to schedule catchups with each of you every 2 weeks - go to the appointment page

<https://calendar.google.com/calendar/selfsched?sstoken=UUpQWXVKcW1fWGhkfGRIZmF1bHR8MTg1YTMzNGM0NmM4NDQzN2JhZjNIZDczOTgzMjkxY2Y> and book an appointment for either 25 or 26 June.

Thursday 11 June - Week 9 Term 2

Appointments with me

I need to see everyone to discuss your progress and what next. I want to do this tomorrow - please book an appointment slot with me at

<https://calendar.google.com/calendar/selfsched?sstoken=UUxDSUtdTRiNk5CfGRIZmF1bHR8OWYwMDI5ZTZIMWZkZWRIYmViMTc3NDImMzBjMjgzZjQ>

I am writing reports next week -so good progress will help me write a good report!

There is a section in the UX checklist about relevant implications. This is easy, once you have thought it through. Relevant implications also appear in the programming assessment and one of the externals (for 2020

- legal
- sustainability and
- health and safety.

), so it is worth understanding these well. They don't need a lot of work, just a bit of thinking, and a small bit of writing.

Programming assessment.

We will start this properly today.

The assessment is here: https://drive.google.com/file/d/1VQvgP63L6KIKQ2xFQHFy32ISaWby_gle/view

Key action points:

- Read the assessment from front to back including the marking rubric
- Get your brief “tightened up” with a very clear list of **essential features** and ‘**nice to have features**’ these need to be bullet pointed.
- Have a long, hard think about what your GUI and data storage will be like - in broad terms. This is what we will talk about in the student catchups tomorrow.
- Key points:
 - Work needs to be in Github
 - Need to use Github projects or trello
 - Every week we/you will do a 5 minute standup, saying what you done and take questions from me and other students
 - The log is a key document and needs to be updated weekly

Thursday 21 May - Week 6 Term 2

Appointments with me

Well last week was a bit disastrous with my internet (I live rurally, and the point to point wireless is solar/wind powered - so weather is needed to power it). I still need to see:

~~Arlø~~

~~Mikael~~

~~Marshall~~

~~Matthew~~

Asap - please give me a time tomorrow that works.

We are spending today going through the assessments and planning your next steps.

To do:

- Read UX assessment at least twice!
- Do the research on UX methodologies (1 page)
- 2 or more rough ideas for your GUI - develop on paper/or digital
- All go into the shared folder

Thursday 14 May - Week 5 Term 2

Appointments with me

I need to have a conversation with each of you about your project - either today or tomorrow afternoon. Please book it at

<https://calendar.google.com/calendar/selfsched?sstoken=UUpQWXVKcW1fWGhkfGRIZmF1bHR8MTg1YTMzNGM0NmM4NDQzN2JhZjNlZDczOTgzMjkxY2Y>

Bits and Pieces:

There are a couple of things we need to do before we can start the project/assessments:

- Have an understanding of how to manage our project/be able to manage our project
- Have an understanding of how to manage our code/be able to manage our code

Notice how these are two different things.

The project is about deciding what to do now and next, consulting with your stakeholder, getting feedback and ensuring your project meets the requirements and is on time.

The code is about ensuring your code is well documented, stored safely and that different versions can be saved using proper tools.

Project Management

The key idea here is to break your project down into very small parts, make a single part, test it with your stakeholder, get their feedback, act on it, make some more and improve, test again, get feedback, make & improve.... Until the project is completed. Note how you DO NOT finish your project and then give it to your stakeholder for the first time. You should aim to have at least 4 cycles of development with your stakeholder - and meet with them every 2 weeks. This is called an agile development method, (agile because it responds to the stakeholder feedback right through the development process)

The kanban board (explained below) is a really good way to help you manage the development process. It can be done using either Trello or Github projects (GHP). Which you choose is up to you. Trello is a totally separate product, GHP is integrated into Github which you will use. Either way, using project management tools is required for the project standard.

- Using either github projects, or trello, [make a kanban board \(and here\)](#)
 - [This page has easy information](#) on how to use Trello to manage your project. You will need to be familiar with how to use Trello - it will be part of your assessment
 - There are also [lots of videos](#) on how to use Trello for managing projects.
 - Github projects is an alternative to Trello. Read about it here <https://help.github.com/articles/about-project-boards/> and in more detail here <https://help.github.com/articles/tracking-the-progress-of-your-work-with-project-boards/>

Managing Code

By this we mean your code is well documented, stored safely and that different versions can be saved using proper tools. You will use Github. GH can be integrated with both netbeans and IntelliJ. Your github username should be on here

https://docs.google.com/spreadsheets/d/1-ZMEI8eCkD27Qke-bW47g0_QbXRYRdbLQjLtEbvLJJc/edit#gid=0

The two concepts that can be confusing with Github is that of local and remote repositories.

Local is stored on your computer/laptop.

Remote is a copy of the local, but stored on github's servers. You need a local copy to push it to the remote. The remote is more secure, because it is a backup of your local.

Additionally you can copy a remote repository (a clone of it) to your local and then push it back to remote with changes!

Using Github with IntelliJ is outlined at

Thursday 7 May - Week 4 Term 2

Some more JavaFX tutorials:

- Here's a series of using JavaFX/IntelliJ tutorials that look very useful <https://www.youtube.com/playlist?list=PL6gx4Cwl9DGBzfXLWLSYVy8EbTdpGbUIG> I'm sure you will have had to do the tutorial from last week to make sense of this, and I have not tried these out, but they look extremely useful.
- And here's a non-video resource <https://www.tutorialspoint.com/javafx/index.htm> (Menu on the left or PDF download (\$))

And some Swing ones:

- One I wrote myself <https://docs.google.com/document/d/1V6fu7FggvGMkJv-hkq8E1IZAY9LVPnEQhPKYrSWrWt4/edit#heading=h.s65zxjnomg6u> **Mikael & Dakin do this one second**
- <https://www.youtube.com/watch?v=GZ9MT2myBf8> This is a long tutorial but covers a lot of Swing. You will have to do the simple Netbeans tutorial from last week. **Mikael & Dakin do this one third**

To Do:

- Ensure you have done the JavaFX and Swing stuff from last week.
- Have a look at the Swing/JavaFX resources above. You are probably not yet at the point where you can choose which to use for your project as it is yet undecided.
- Project ideas: Now is the time that we need to firm up at least 2 ideas for a project. [This document](#) outlines the scope of the project.
- Complete [this document that outlines your ideas](#)
- On paper, sketch what the GUI would look like for both ideas - this will really help you understand what the projects could look like. At this point you will not have contacted your potential stakeholder yet. That's

Thursday 30 April - Week 3 Term 2

GUI's

Let's take a look at GUI's. Having a graphical user interface (GUI) is likely to be a key part of your project.

To build a GUI you will need to know:

- How to build a GUI (the relevant tools and techniques)
- How to make the GUI interact with the programme backend (i.e. the data)
- How to plan and design a usable GUI, so that it is easy for users to interact with (This is covered in AS 91901 Apply user experience methodologies to develop a design for a digital technologies outcome 3 credits Internal)
- How to test a GUI - linked in with the above AS

That is a lot to take in! GUI's work on the basis of something called "Event-driven" programming which works as follows:

User interacts with the GUI in some way - an event e.g.:

Presses a button
Uses a drop down list
Drags something
Adds text to a text box
etc



The interaction fires a "trigger" in the program that does something such as:

stores/modifies data
Changes something on the GUI
Etc

We will look at:

- How to build a GUI (the relevant tools and techniques)
- How to make the GUI interact with the programme backend (i.e. the data)

Together.

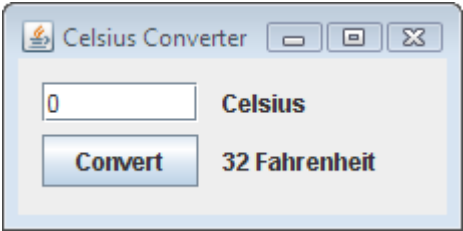
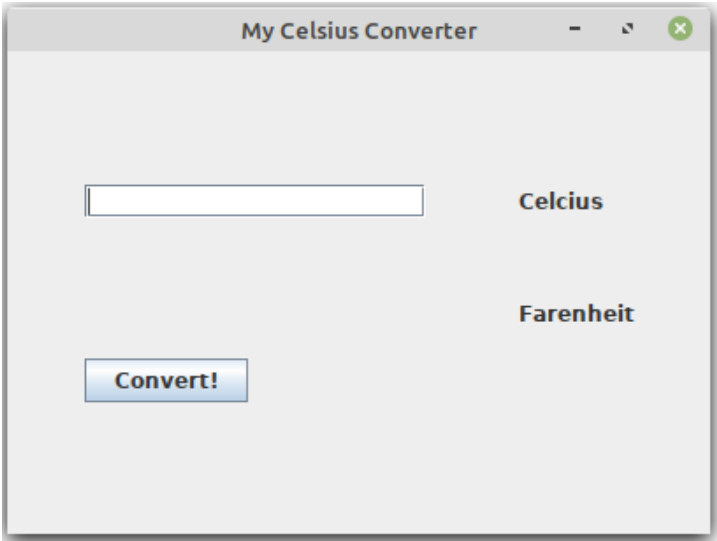
How to build GUI's that interact with the program backend

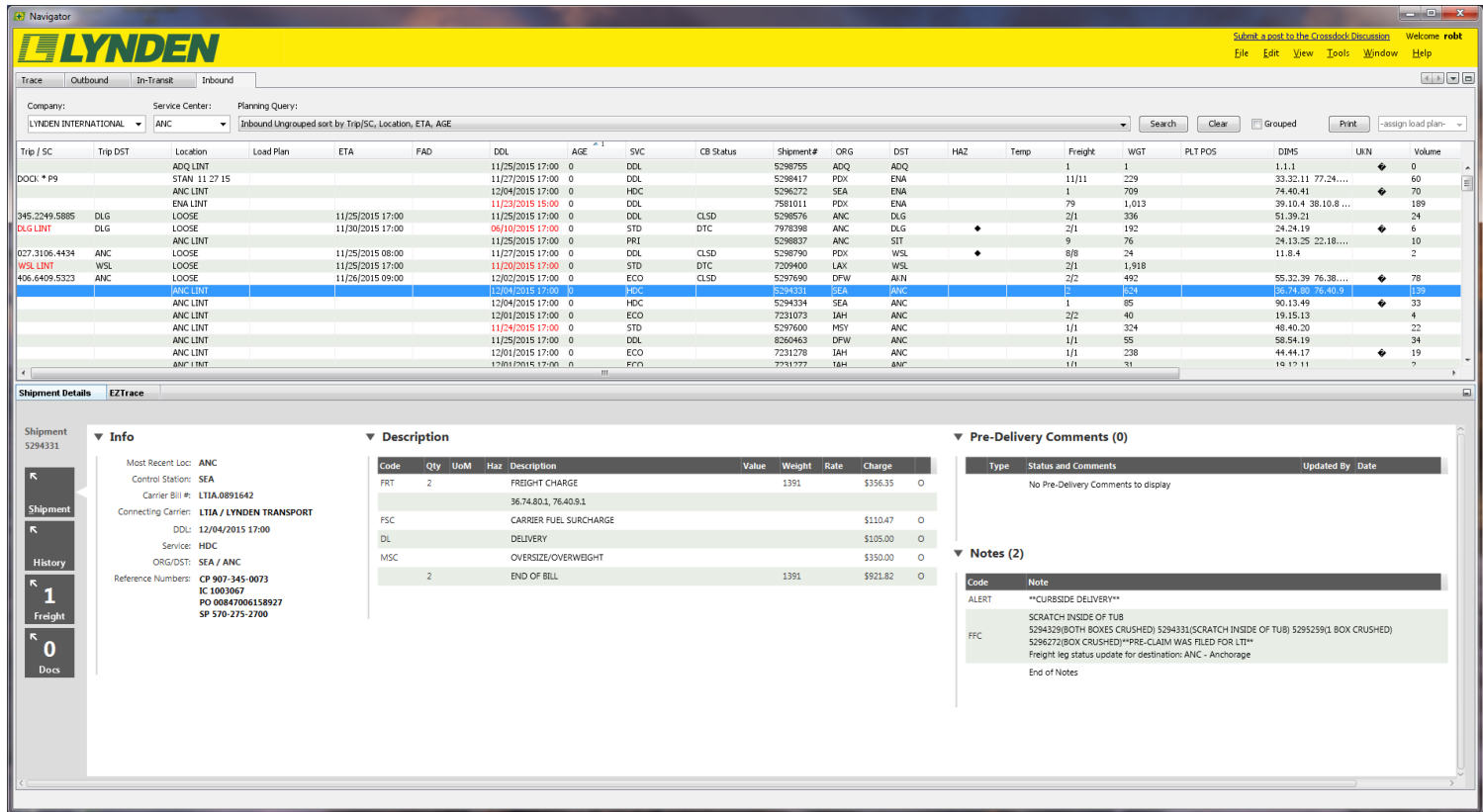
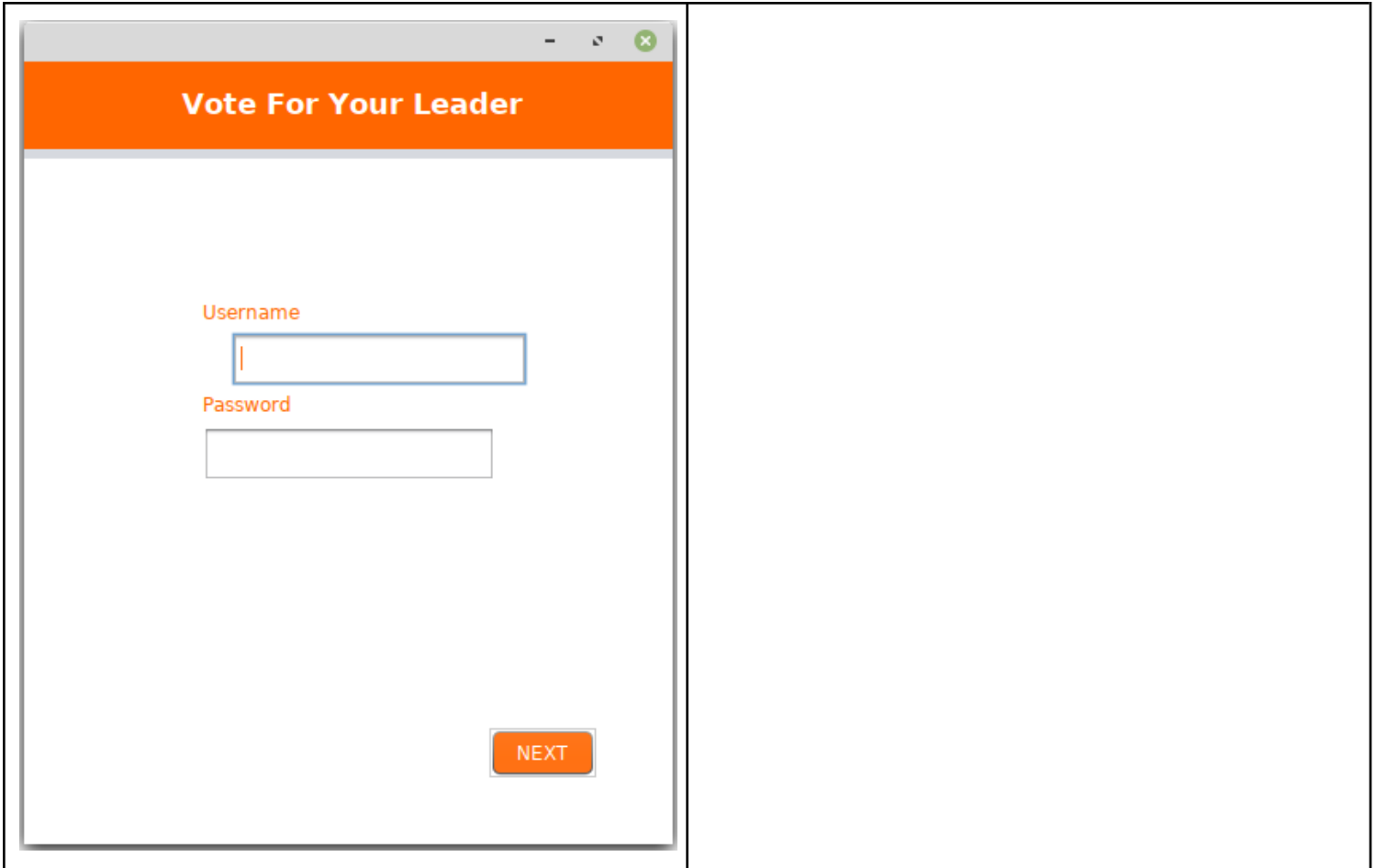
Building GUI's can be done either by completely hand-coding everything or using a drag and drop plugin to an IDE such as Netbeans or IntelliJ IDEA. Either can make great GUI's, drag and drop is usually faster, but means there is some code you won't understand (that's OK as long as you understand the important parts). In Java there are two different types of GUI that I can support you with, Swing and JavaFX. This means we have 4 combinations of 2 things we need to consider Netbeans or IntelliJ IDEA and, Swing or JavaFX? Which one to choose?

	Netbeans	IntelliJ
Swing	Great support with a built in GUI designer	OK support with a plugin GUI designer
JavaFX	New versions have poor support	Good support with a plugin GUI designer

If I was making a Swing GUI I would use Netbeans (unless I was very familiar with IntelliJ) and IntelliJ if I was making a JavaFX GUI.

Java Swing vs JavaFX

Java Swing	JavaFX
Mature (lots of existing code/tutorials and GUI components) Much older look and feel 	Much younger (fewer tutorials/ examples/ GUI components) Modern look and feel Can be styled using CSS 



The top area is in Swing, the bottom in JavaFX.

To Do

To Do:

This assumes you have installed both Netbeans and IntelliJ.

You need to make a simple temperature converter (converts Farenheit into Celcius) using both Netbeans **Swing** and IntelliJ **JavaFX** so you can compare the two. **Do the Netbeans one first.**

<https://docs.oracle.com/javase/tutorial/uiswing/learn/> Mikael & Dakin do this first

Then the IntelliJ one - This is a bit more involved to get started with so there are quite a few more steps to start off with.

Instructions to get started using IntelliJ IDEA <https://www.jetbrains.com/help/idea/javafx.html#>

- I followed these and they worked.
- Using scene builder initial configuration
<https://www.jetbrains.com/help/idea/opening-fxml-files-in-javafx-scene-builder.html#> (I needed to download the scenebuilder jar when prompted from IDEA when opening the fxml file).
- Develop a simple application also works
<https://www.jetbrains.com/help/idea/developing-a-javafx-application-examples.html#>
- **Do the Netbeans C to F converter using IntelliJ** <https://www.youtube.com/watch?v=-SmNpKskfJc>

Other GUI tutorials to do

We also now need to start thinking about the project that you will do this Term:

I have written a document about this

<https://docs.google.com/document/d/1J8Op8z-0D1kCQeCRaHb6mGIVgYFZaAgUDcD8ym6sAY8/edit>

Please read through it. Do any projects come to mind?

Thursday 23 April - Week 2 Term 2

There are a few more things we need to cover: You'll notice that we are touching on each technique or strategy quite lightly. That's because your project in Term 2 can use a range of techniques. It's important for you to know the range of strategies that can be used - and where to find help for them. It's less important that you know each strategy in detail yet - if you use it in your project you will have plenty of time to learn it.

Last week we looked at custom classes - the pen class. Data was entered when the program was running and saved using the serialization method - a useful way to store the state of a program. The problem with this technique is that all the data must be typed in by hand.

What if you had a spreadsheet of data to load? In this case it would be better to load the spreadsheet. We can do this by reading the spreadsheet (where the values are separated by commas (',')) and saved as a csv (comma separated values) or text file. Once the data is read we can create an arraylist of pen objects from the data.

I have broken this down into logical steps. **This is covered in detail in the Otago Uni Level 3 workbook - chapter 5** - the examples were taken from there and modified - I suggest you look at these alongside the repls below.

1. <https://repl.it/@trevorstorr/read-from-a-file>

This reads data from a file - data.txt

2. <https://repl.it/@trevorstorr/read-from-a-file-1>

This reads data from a file, puts each line of data (make, model, type, colour, price) as an element in an arraylist and reads the arraylist

3. <https://repl.it/@trevorstorr/read-from-a-file-2>

This is the same as #2 but split into separate methods for reading the file and printing the arraylist contents - with a main method to control it all - reads the data into an arraylist

4. <https://repl.it/@trevorstorr/read-from-a-file-3>

This builds on #3 so that we can access for example the colour or price of a pen separately, rather than all at once - but still as items in an arraylist - not objects of class Pen.

5. <https://repl.it/@trevorstorr/read-from-a-file-4>

This is almost exactly the same as #4 - but the data.txt file has a header line so you can be clear what each item is

6. <https://repl.it/@trevorstorr/read-from-a-file-5>

This develops the previous examples by reading the data file and making an arraylist of pen objects (using the Pen class from last week)

Task - work through the above examples so that you understand them.

Task - Create a fork of [this pen repl](#) - Modify it so that pens can be loaded from a txt file. You will need to modify the menu so that it can be controlled - ie add an option to load/import from the txt file. If you do this correctly, you should be able to load pens from the data.txt file and add additional pens from the text interface. - How will you create an import type functionality that does not overwrite any existing pens in the arraylist?

Task - add the ability to load from a file to the program you created last week - the garage program.

6. <https://repl.it/@trevorstorr/create-new-file-and-write-to-it>

Create a new text file and write to it.

7. <https://repl.it/@trevorstorr/append-text-to-an-existing-text-file>

This takes an existing file "names.txt" which has two names in it and adds several more names to the file. There is quite a nice explanation of how this works here:

<https://stackoverflow.com/questions/1625234/how-to-append-text-to-an-existing-file-in-java#1625263>

Hashmaps

Hashmaps are a type of data structure that allows you to store information as a pair. For example name:address or NSN number: Name

It is much simpler to use hashmaps for data that naturally occurs in pairs than an array or arraylist.

There are special methods for inserting, deleting and searching hashmaps, and also specialised types of hashmaps.

[This video introduces the concept of hashmaps](#) - sometimes called hash tables (the last part on linking is not required at this level)

[This page has a really simple introduction](#) to hashmaps in Java.

And [here is a more detailed page](#) that has an example and documentation on how to use hashmaps.

I have made a simple repl that demonstrates this for NSN numbers and student names.

<https://repl.it/@trevorstorr/Hashmap-example>

Task - copy my repl into one of your own. Use the iterator in the “more detailed page” example to find the value “Trevor Storr” and get the NSN number for that entry - ie find the key for a value.

Thursday 16 April - Week 1 Term 2

Outline of Term 2:

- Classes (again)
- Reading to and from files
- GUI's (netbeans / intelliJ)
- Github/Trello
- Your project

This week we will re -visit classes. In particular we will look at how making custom -ones designed especially for the program in question - classes can be used to ensure data is kept safe. Next week we will take a look at reading and writing from text files. After that there will be some stuff on GUIs - and then your project.

The repl <https://repl.it/@trevorstorr/classes-pens-1> has 2 classes Main and pens. Watch the video (i'll put a link in when I've made it) explaining the purpose and organization of the program. In particular, ensure you understand the concept of **setter** and **getter** methods for the pen class. The pen objects are held in an arraylist. Therefore the arraylist has elements all of type pen that each contain details of each individual pen.

Task - write a similar program for a garage that has cars. - Do this in repl. - share it with me

Repl pairs(only if you want to)

<https://docs.google.com/spreadsheets/d/1ywde6RxqH6IMu4IKP1xJ7pGCtaqGcloMk3mnRE29OUY/edit#gid=0>

At the moment, whenever we create instances of custom classes, the data for each class - e.g. pen colour etc is manually coded into the program. This is not scalable for bigger programs - we need some way to save and load our data.

One way is to read and write data to a text or comma separated value (CSV) file. We will look at this a bit later - next week perhaps.

In the meantime, we will look at a technique called serialization. This is where the 'state' of a program is saved as a binary file (you write code to do this). When you want the data from a previous execution of the program, you load the saved serialization file.

Serialization

Serialization is a pretty nifty technique. In short it allows you to save the program state for use at a later point in time. For example, perhaps your program saves data in an array list. Serialization will save your program in a special file that can be loaded so that the data does not have to be re-entered. I have put together a guide on how to use serialization with an example -

https://docs.google.com/document/d/1OGOTp5cEUcA1jM67HFsGlzZRGc_AcpSZWX4rCJbsp8M/edit

An example of this is in a 'pens with serialization' repl.

<https://repl.it/@trevorstorr/Pens-with-menus-and-serialization-1> There are 2 classes - Main and Pen. When you 'save all' you should see a file pens.ser appear. This is created in the method saveAll - about line 128. Loading the file takes place in the loadSaved method - about line 146. There is also a text based menu for the program - this is at the end.

Task - using the example above, add both serialization and a menu to your garage program.

Task - Create another example, this time for books in a library. Have at least 5 fields in the library classes (Title, Author etc....). Add a menu and save data using serialization.

Thursday 26 March - Week 8 Term 1

What a week that was! Fortunately our class can continue more or less, given the present circumstances, as usual. The email I was going to send out last Monday got overtaken by events. I do still need to have a call with each of you. Go to [here](#) and book a 20 minute appointment with me - I am available 9-10am and 1-5pm tomorrow for this.

Last week we looked at indexed data structures and the week before classes - all using repl.it. Repl is good because it is easy for you to use and easy for me to see what you have done. Repl, though, is not very good for doing our big term 2 project on. This is because it cannot make GUI's (graphical user interfaces). To do this we will need to use software that is installed on your laptop or school computer. We also need a way of keeping your work safe.

Not many of you are up to date with the problems - so this is an ideal opportunity to catch up! The total list of essential problems is:

11, 15, 16, 17, 18, 23, 28, 43, 50, 60, 61, 62, 67, 68, 70, 74, 81, 103, 104, 105, 106, 107, 110

Work to do:

- Install [IntelliJ IDEA](#) and [Netbeans](#) on the computer that you use.
- Read through the blurb below and complete the action points

These are development environments that you will use for the Term 2 project (which we will start a few weeks into Term 2). I will post some links on how to use these when I can find updated ones (sometime next week).

This week we are going to have a look at something slightly different - project management tools. In particular we will look at how you can manage a (for you) large project which includes:

- Managing your code
- Planning the project so that it is completed and tested etc

For this I also have in mind your assessment - which will run throughout term 2. This sounds like ages away but it will go very quickly.

The assessment is a project where you have to design, code and manage building an application. You have the opportunity to get 12 credits (split equally between the programming and project parts). I'll explain more about that at the end of this term.

I have selected two tools that I will require you to use for your assessments.

For managing you overall project, You may use Trello or Github projects. There is no real difference between the two. Obviously with github everything is in one place.

To use Trello:

- Create an account on Trello.com. I always use the sign in with Google option as it is more secure and there's one less password to remember.
- Create a team - add me to the team by email lsior@netnz.school.nz
- Put your trello username and email in [this master document](#) (very important)
- [This page has easy information](#) on how to use Trello to manage your project. You will need to be familiar with how to use Trello - it will be part of your assessment
- There are also [lots of videos](#) on how to use Trello for managing projects.

For Git hub projects, see the section on github below.

For managing your code I suggest a combination of 2 oddly named things, netbeans and github.

These are a combination of an integrated development environment (IDE) and a code versioning system.

- IDE - brings together in one place the tools to write code, design GUI's (Graphical Interfaces) and some other things - but can be complicated
- Code versioning system - allows you to save code online and also go back to previous versions of your code. This is essential (your code is safe) but can be a bit complicated. Code is stored in repositories that

can be shared privately. NetNZ has an educational Github account which means I can create private repositories for each of you and share them with the account you create (see below)

Create a Github account

- Go to <https://github.com/> and create an account (top right, sign up) it is essential that you use your real email address, for our purposes it does not matter if the email address is your school one or a private one.
- Once you have completed the signup process, fill in your details (name (done for you), github username, email address) in [this spreadsheet](#). This step is essential because I will share a private github repository (where code is stored) with each of you and so will need your github username/email address.
- I will then share a private Github repository with you. This is where your (remote) code will reside. Only you and I can access it. The repository will be shared via an email link. Github will allow you to synchronise a repository on your work computer (laptop) with the remote computer (github), and I (teacher) will have access to the github version. Accept the invitation.
- Github projects is an alternative to Trello. Read about it here <https://help.github.com/articles/about-project-boards/> and in more detail here <https://help.github.com/articles/tracking-the-progress-of-your-work-with-project-boards/>

[Instructions on using Github](#) (we will have further sessions on this)

Thursday 19 March - Week 7 Term 1

Lesson held at 9am Friday 20 March

I will send out an email re catching up individually with all of you sometime during Monday - Wednesday next week. Email will get to you before Monday - maybe even later today.

Work to do - this includes catching up from last week:

- The material for this week is outlined at https://sites.google.com/netnz.school.nz/csp/l2-revision?authuser=1#h.p_LCZHj6HOxEu5
- The relevant problems are: **70, 74, 81, 103, 104, 105, 106, 107, 110. Only attempt the other problems between 70 and 115 if you are super enthusiastic!**
- Work on the problems from last week if you are not up to date.
- Remember to use the cheatsheet <https://introcs.cs.princeton.edu/java/11cheatsheet/> The heading to look

out for in the cheat sheet is *Functions*

- here is the spreadsheet of problems & resources
<https://docs.google.com/spreadsheets/d/1mV1n1aHrVNi1SGYm5TXpsFfRhKy57G4zzkJaosTWnS0/edit#gid=0>
- Those who are enthusiastic can do week 7!

Thursday 12 March - Week 6 Term 1

Work to do:

- The material for this week is outlined at
https://sites.google.com/netnz.school.nz/csp/l2-revision?authuser=1#h.p_8aSOGqYFMx5X
- Ensure you have completed and submitted problems 61, 62 and 67 & 68
- In my experience, this takes a while to understand - there is **a lot of new ideas** to take in.
- Remember to use the cheatsheet <https://introcs.cs.princeton.edu/java/11cheatsheet/> The heading to look out for in the cheat sheet is *Functions*
- here is the spreadsheet of problems & resources
<https://docs.google.com/spreadsheets/d/1mV1n1aHrVNi1SGYm5TXpsFfRhKy57G4zzkJaosTWnS0/edit#gid=0>
- Those who are enthusiastic can do week 6!

Thursday 5 March - Week 5 Term 1

Reminder - any problems doing the problems, use the group forum, or ask for an extra tutorial. I will contact your eDean after tomorrow's tutorial if you are behind and have not contacted me separately to get some extra help to catch up! 23 and 28 should have been completed - ignore the first two rows they are my accounts.

22	23	24	25	26	27	28
✗	✗	✗	✗	✗	✗	✗
✗	✗	✗	✗	✗	✗	✗
✗	✗	✗	✗	✗	✗	✗
✗	✗	✗	✓	✗	✗	✗
✓	✓	✓	✓	✗	✗	✗
✗	✗	✗	✗	✗	✗	✗
✗	✗	✗	✗	✗	✗	✗
✗	✗	✗	✓	✗	✗	✗
✓	✓	✓	✓	✓	✓	✓
✓	✗	✗	✓	✗	✓	✗

If you need help with week 2 & 3 - please, please, please let me know!

Work to do:

- The material for this week is outlined at https://sites.google.com/netnz.school.nz/csp/l2-revision?authuser=1#h.p_KN6VU-g73lbH

- Ensure you have completed and submitted problems 43, 50 & 60
- Remember to use the cheatsheet <https://introcs.cs.princeton.edu/java/11cheatsheet/> The heading to look out for in the cheat sheet is *Functions*
- here is the spreadsheet of problems & resources
<https://docs.google.com/spreadsheets/d/1mV1n1aHrVNi1SGYm5TXpsFfRhKy57G4zzkJaosTWnS0/edit#gid=0>
- Those who are enthusiastic can do week 5!

Thursday 27 Feb - Week 4 Term 1

(Actual lesson - 9am Friday 21) Reminder - any problems doing the problems, use the group forum, or ask for an extra tutorial

If you need help with week 1 & 2 - please, please, please let me know!

Work to do:

- The material for this week is outlined at
https://sites.google.com/netnz.school.nz/csp/l2-revision?authuser=1#h.p_6YFiMm3zEogp
- Ensure you have completed and submitted problems 23 & 28
- Remember to use the cheatsheet <https://introcs.cs.princeton.edu/java/11cheatsheet/> The 3 headings to look out for in the cheat sheet are *Anatomy of a while loop*, *Anatomy of a for loop*, *loops*, *Break statement*, *Do-while loop* and *Switch statement*
- here is the spreadsheet of problems & resources
<https://docs.google.com/spreadsheets/d/1mV1n1aHrVNi1SGYm5TXpsFfRhKy57G4zzkJaosTWnS0/edit#gid=0>
- Those who are enthusiastic can do week 4!

Thursday 20 Feb - Week 3 Term 1

(Actual lesson - 9am Friday 21) Reminder - any problems doing the problems, use the group forum, or ask for an extra tutorial

If you need help with week 1 - please, please, please let me know!

Work to do:

- The material for this week is outlined at https://sites.google.com/netnz.school.nz/csp/l2-revision?authuser=1#h.p_rdM07PdVzCow
- Ensure you have completed and submitted problems 15, 16, 17 & 18
- Remember to use the cheatsheet <https://introcs.cs.princeton.edu/java/11cheatsheet/> The 3 headings to look out for are Anatomy of an if statement, If and if-else statement & Nested if-else statement.
- , the y12 revision book & htllacs here is the spreadsheet of problems & resources <https://docs.google.com/spreadsheets/d/1mV1n1aHrVNi1SGYm5TXpsFfRhKy57G4zzkJaosTWnS0/edit#gid=0>
- Those who are enthusiastic can do week 3!

Thursday 13 Feb - Week 2 Term 1

Reminder - any problems doing the problems, use the group forum, or ask for an extra tutorial

Work to do:

- Ensure you have accepted invites from repl.it & can get into it
- Check you have got the google group link (for help/discussion) netnzcsp2020@googlegroups.com
- The material for this week is outlined at <https://sites.google.com/netnz.school.nz/csp/l2-revision>
- Ensure you have completed and submitted problem #11 String Methods practice 2 (though it is quite likely you will need to do 1 -10 to be able to do this.
- Remember to use the cheatsheet <https://introcs.cs.princeton.edu/java/11cheatsheet/> (Strings are nearly at the end) , the y12 revision book & htllacs here is the spreadsheet of problems & resources <https://docs.google.com/spreadsheets/d/1mV1n1aHrVNi1SGYm5TXpsFfRhKy57G4zzkJaosTWnS0/edit#gid=0>
- Those who are enthusiastic can do week 2!

Before Thursday 13 Feb - Week 1 Term 1

Kia ora welcome to the course!

I am really looking forward to working with you this year. You can find a little bit more about me and the course on our course site - see the link above.

- Find the home menu
- watch the videos about me and the course.
- Under the software menu, follow the instructions to create a repl.it account

Next - the challenging part!

It's really important that I know a little bit about you - in previous years I have gotten learners to complete a questionnaire, but I think there are more effective ways of doing this. Your job is to create a video about yourself that answers or explains the following points:

- All the usual stuff about name, school etc
- What your experience of digital technologies is - what have you done in previous years (details please!) did you enjoy it - what parts did you find difficult etc - do you do any programming out of school. I need to get a real flavour of your ability and interest in this subject. Don't worry if this is new to you - many students have done very well with minimal to no prior experience.
- What challenges you might have completing the course - this could range from what you think your weaknesses are, through to specific learning difficulties and work or sporting commitments - be honest!
- What results you got at NCEA level 2 - especially Maths, English and Digital Technologies - the rough number of credits and grades will do here (e.g. English 20 credits of which 15 were Merit and 5 Achieved)
- What you hope to do after NCEA level 3 - if it is going on to Tertiary say which subject interests you at this point in time.
- What your hobbies and interests are (useful for me to know when we are thinking about projects)

The video should be no shorter than 5 minutes - and no longer than 10.

Some tips:

- Hold the phone horizontally to shoot in landscape
- Have the notes above so that you can see them - tackle the points in order
- Audio and lighting are important - speak clearly and slowly - imagine you are talking to a friend. I can be useful to use earphones with a built in microphone
- Don't write down or script the answers - this is not a test!

- I am not looking for or expect these videos to be 'polished' they are simply so that I can find out a little bit about you!

Once you have finished the video, upload to youtube, edit the beginning/end to clip to the relevant part and save unlisted, so it is not searchable. This means that only people who have the url will be able to see it.

Send me the URL (tstorr@netnz.school.nz) - so that I can see it!

I look forward to seeing you all!

Trevor