LETTER 1

Let’s get to know each other
IN THIS LECTURE

TODAY....

- Welcome and Introductions
- Course Administration
- How COMP1511 works
- How to get help and the best ways to approach learning Programming
- What is programming?
- What is Linux and working in Linux
- A first look at C
WHO AM I?

JAX
Teaching Assistant
Loves long walks, treaties and pats

DR SASHA VASSAR
Lecturer in Charge/Course Convenor
Loves dogs, teaching, solving complex problems and having a good yarn...

JUNO
Teaching Assistant
Loves sleeping in random places
Admin Extraordinaire
Has a script for everything

Shrey Somaiya
Admin Extraordinaire
Always awake

Ben Briant
Admin Extraordinaire
Forum king
THE WONDERFUL TUTORING TEAM

https://cgi.cse.unsw.edu.au/~cs1511/22T1/team/
All course information can be found HERE (not Moodle!)

https://cgi.cse.unsw.edu.au/~cs1511/22T1/
EMAIL THE COURSE EMAIL FOR ALL ADMIN RELATED ENQUIRIES:

cs1511@cse.unsw.edu.au

ADMIN RELATED ISSUES:

FOR ANY ENROLMENT ISSUES:

UNSW Nucleus Student Hub
https://nucleus.unsw.edu.au/en/contact-us

ELP PLANS

If you have an ELP plan in place, please email me directly on

a.vassar@unsw.edu.au
FORUM

Post all your questions here and feel free to answer other's questions
https://edstem.org/au/courses/7587/discussion/

ASK QUESTIONS IN TUT/LABS

HELP SESSIONS

Schedule will be announced shortly
Link to our Hale FastPass system will be provided shortly

Good place to get help outside of normal lab/tutorial times
It is your intro to programming
This is where the journey starts :)

Computers can only follow instructions that we give them to solve problems
Writing a program is providing the computer with a set of instructions
Problem solving is a very important skill, can only be built up with practice!
• We assume no prior knowledge & zero previous programming experience
• We teach you the fundamentals of programming, how to approach and solve problems, and how to talk to computers in a common language
• Streamed online via YouTube Live (recordings will be available)
  ○ Tuesday 4pm-6pm (AEDT)
  ○ Friday 3pm-5pm (AEDT)
• Week 6 is Flex Week, so no formal lectures but we have a very exciting series of lectures on Ethics going!
• If you have a question, feel free to ask in live chat,
• Please be respectful of others at all times - everyone is here to learn.
• Theory - What are we trying to understand?
• Demonstrations - Some live coding to show you how some things work
• Problem Solving - How do we decide what to code?
• Other stuff - Outside of programming, what’s important?
• Lecture slides (and other materials) are available from the Course Website https://cgi.cse.unsw.edu.au/COMP1511/22T1/
• Lecture recordings will be in the YouTube playlist and linked via the Course Website
Go further in depth into the topics we're teaching
Actual practical working of tasks and problems we've given you
Learning how to solve problems before you write the code!
Tutorial Questions will be available in advance of the tutorials on the course website: https://cgi.cse.unsw.edu.au/COMP1511/22T1/
"Tutorials are a good place for interactive learning. You’ll have time to discuss and work through problems there."

- Online and face-to-face: please check your timetable for your enrolment details
  - For online classes, use Blackboard Collaborate (link on Moodle)
  - Please turn on your cameras if you can
  - We love seeing pets make an appearance
- Sample answers released after the last tutorial for the week
Practical coding including working in small groups
Time to have one on one conversations with your tutors
Problem sets will be marked automatically and count towards your final marks (15% total over the term)
There are challenge exercises for earning bonus marks (not necessary and some are hard enough that they'll eat up a lot of time)
Tutorials and Labs do NOT run in Week 6
Individual work

These will take you a few weeks and will test how well you can apply the theory you've learnt.

There are three Assignments due:

- Assignment 0 - 5% (Monday 8pm Week 4)
- Assignment 1 - 15% (Monday 8pm Week 7)
- Assignment 2 - 25% (Friday 8pm Week 10)

Late penalties of 1% per hour apply (this reduces your maximum possible mark).

"Start the assignments early, so that you have time to chip away and get help as needed."

- Individual work
- These will take you a few weeks and will test how well you can apply the theory you've learnt
- There are three Assignments due:
  - Assignment 0 - 5% (Monday 8pm Week 4)
  - Assignment 1 - 15% (Monday 8pm Week 7)
  - Assignment 2 - 25% (Friday 8pm Week 10)
- Late penalties of 1% per hour apply (this reduces your maximum possible mark)
HELP SESSIONS
OPTIONAL SESSIONS SCHEDULED DURING THE WEEK

“"A great place to ask questions and get help to fill any gaps.”

- Held using Blackboard Collaborate (you can find the link on Moodle)
- Some one on one consultation with tutors
- Time for you to ask individual questions or get help with specific problems
- Schedule will be up on the Course Website soon
- These are particularly busy around Assignment deadlines
- Trialling Hale - a FastPass Booking System to register for a ticket to attend a help session
Expected workload of around 3-5 hours total
You’ll be given a series of problems to solve in C
You will also be expected to read some C and show you understand it
There will also be some questions covering programming ideas

Exam Hurdles
- Parts of the exam are competency hurdles
- These questions must be answered correctly to pass the course
To pass the course you must:

- Score at least 50/100 overall
- Solve problems using arrays in the final exam
- Solve problems using linked lists in the final exam
Special Consideration:

- Support for any issues that make it difficult for you to study
- https://student.unsw.edu.au/special-consideration
- You can apply now if you have existing reasons (or later if something comes up)

If you have an ELP plan, please email it directly to me:

a.vassar@unsw.edu.au
A Supplementary exam can be offered to students granted Special Consideration for the exam.

- Fit-to-Sit rule
- Identical in format to the main exam
- Held sometime in the period 23-27 May, so you must make yourself available if you have been granted a supplementary exam.
This course and this University allows all students to learn, regardless of background or situation. Remember the one rule . . . you will not hinder anyone else’s learning!

Anything connected to COMP1511, including social media, will follow respectful behaviour:

- No discrimination of any kind
- No inappropriate behaviour
  - No harassment, bullying, aggression or sexual harassment
- Full respect for the privacy of others
"If you don't spend the time to learn and practice the content, the only person who loses is you."

- Plagiarism is the presentation of someone else's work or ideas as if they were your own.
- Any kind of cheating on your work for this course will incur penalties (see the course outline for details)
- Collaboration on individual assessments like Assignments is considered plagiarism
The internet has a lot of resources you should learn to use, just make sure you credit your sources.

No collaboration at all on individual assignments.

Your submissions are entirely your own work.

Don't use other people's code.

Don't ask someone else to solve problems for you (even verbally).

Don't provide your code to other people.

"Discussion of work and algorithms is fine (and encouraged)."
At best, you'll lose the marks for the particular assignment
At worst, you'll be asked to leave UNSW
And even worse . . . you won't learn what you paid all this money and time to learn
• Course webpage
• Course forum
• Recorded Lectures (replay YouTube Streams or via Moodle)
• One on One
  ◦ Ask your tutor during lab sessions
  ◦ Help Sessions
• Serious Issues
  ◦ Email: cs1511@cse.unsw.edu.au
  ◦ The Nucleus: nucleus.unsw.edu.au
  ◦ CSE Help Desk: http://www.cse.unsw.edu.au/~helpdesk/
<table>
<thead>
<tr>
<th>Student Support</th>
<th>I Need Help With...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>My Feelings and Mental Health</strong></td>
<td>Managing Low Mood, Unusual Feelings &amp; Depression</td>
</tr>
<tr>
<td>Mental Health Connect</td>
<td>student.unsw.edu.au/counselling Telehealth</td>
</tr>
<tr>
<td>Mind HUB</td>
<td>student.unsw.edu.au/mind-hub Online Self-Help Resources</td>
</tr>
<tr>
<td><strong>In Australia Call Afterhours</strong></td>
<td>UNSW Mental Health Support Line 1300 787 026 5pm-9am</td>
</tr>
<tr>
<td><strong>Outside Australia Afterhours</strong></td>
<td>24-hour Medibank Hotline +61 (2) 8905 0307</td>
</tr>
<tr>
<td><strong>Uni and Life Pressures</strong></td>
<td>Stress, Financial, Visas, Accommodation &amp; More</td>
</tr>
<tr>
<td>Student Support</td>
<td>Indigenous Student Support</td>
</tr>
<tr>
<td></td>
<td>student.unsw.edu.au/advisors</td>
</tr>
<tr>
<td></td>
<td>nura-gili-centre-indigenous-programs</td>
</tr>
<tr>
<td><strong>Equity Diversity and Inclusion (EDI)</strong></td>
<td>student.unsw.edu.au/sexual-misconduct</td>
</tr>
<tr>
<td><strong>Educational Adjustments</strong></td>
<td>To Manage my Studies and Disability / Health Condition</td>
</tr>
<tr>
<td>Equitable Learning Services (ELS)</td>
<td>student.unsw.edu.au/els</td>
</tr>
<tr>
<td><strong>Academic and Study Skills</strong></td>
<td></td>
</tr>
<tr>
<td>Academic Skills</td>
<td>student.unsw.edu.au/skills</td>
</tr>
<tr>
<td><strong>Special Consideration</strong></td>
<td>Because Life Impacts our Studies and Exams</td>
</tr>
<tr>
<td>Special Consideration</td>
<td>student.unsw.edu.au/special-consideration</td>
</tr>
</tbody>
</table>
LEARNING IS HARD...

"Learning programming is a secondary skill (like many others!) — it is not intuitive like learning how to speak..."

Secondary skills are learnt slowly and with conscious and deliberate effort. It is not magic and it will not happen overnight, you have to keep practising and building up your knowledge base. Don't feel disheartened if you do not understand something first go - try and try again, get help, let us know if there is something that is just not making sense. Make sure to attempt all your labs questions and assignments, working through these problems will help you build an understanding of how to solve similar problems, and how to use code to solve these.
Suppose you have two buckets. One of these is 3L in capacity and the other one is a 5L bucket. How could you measure exactly 4L using only those buckets and as much extra water as you need?
WHAT IS A COMPUTER?
A TOOL . . . A MACHINE . . .
THE LOVE OF MY LIFE...

The ultimate tool in its ability to be reconfigured for different purposes.

The key elements:
- A processor to execute commands
- Memory to store information

Some trivia:
WHAT IS PROGRAMMING?

- Providing a computer with specific instructions to solve various problems
  - Using specific languages to write those instructions (code)

- At the core of it - problem solving!
  - You may go through many iterations before you get it right
  - Mistakes are good!
WHAT IS AN OPERATING SYSTEM?

- An Operating System is the interface between the user and the computer hardware
- Operating Systems:
  - Execute user programs and make solving problems easier
  - Make the computer system convenient to use
- Basically, an Operating System sits between our code and the computer, providing essential services
WHAT IS LINUX?

- Linux is a Unix-based operating system:
  - Open source
  - More reliable
  - Lightweight
  - Faster, and
  - More secure
WHAT IS TERMINAL?

- Terminal (command line driven) allow us to send simple text commands to our computer to help navigate directories, copy files, etc.
- Back in the day this was the only way to communicate with the computer!
- A Terminal is the main interface to Linux
- This means all our interaction is in text...
SOME IMPORTANT TERMINAL COMMANDS

- Lists all the files in the current directory:
  `ls`
- Makes a new directory called `directoryName`:
  `mkdir directoryName`
- Changes the current directory to `directoryName`:
  `cd directoryName`
- Moves up one level of directories (one folder level):
  `cd ..`
- Tells you where you are in the directory structure at the moment:
  `pwd`
File operations on the command line

- Copy a file from the source to the destination
  \texttt{cp source destination}
- Move a file from the source to the destination (can also be used to rename)
  \texttt{mv source destination}
- Remove a file (delete)
  \texttt{rm filename}

The \texttt{-r} tag can be added to \texttt{cp} or \texttt{rm} commands to recursively go through a directory and perform the command on all the files

\texttt{cp -r COMP1511 COMP1511\_backup}

(will copy all files from my COMP1511 directory to my COMP1511\_backup directory)
Our labs are running Linux with the basic tools necessary to get started

You will definitely want to get your own computer ready to code with:

- VLAB allows you to remotely use CSE’s resources - instructions on setting this up available in the first laboratory
- There are other more advanced options that we can help you with also - check the Home Computing site or the guides on our course website
For COMP1511 we need:

- A text editor (like gedit)
  - Helps out a little by highlighting C in different colours (after you run 1511 setup command in lab01)
- A compiler (we use dcc)
  - A translator that takes our formal human readable C and turns it into the actual machine readable program
  - The result of the compiler is a program we can "run"
- You can use VLAB to access CSE's editor and compiler
We need a shared language to be able to have this conversation.

Well be looking at one particular language, C and learning how to write it. C is:

- A clear language with defined rules so that nothing we write in it is ambiguous.
- Many modern programming languages are based on C.
- A good starting point for learning how to control a computer from its roots.
LET'S C SOME C

SORRY CAN'T HELP MYSELF!

```c
#include <stdio.h>

int main (void) {
    printf("Hey!\n");
    return 0;
}
```

// A demo program showing output in C
// Sasha Vassar, February 2022
Hey!
BREAKING IT DOWN INTO PARTS

HEADER (LINES 1 & 2)

- Words for humans
- Half our code is for the machine, the other half is for humans! (roughly)
- We put “comments” in to describe to our future selves or our colleagues what we intended for this code
- // in front of a line makes it a comment
- If we use /* and */ everything between them will be comments
- The compiler will ignore comments, so they don’t have to be proper code

```c
// A demo program showing output in C
// Sasha Vassar, February 2022 Hey!
```
BREAKING IT DOWN INTO PARTS

#include is a special tag for our compiler (line 4)

- It asks the compiler to grab another file of code and add it to ours.
- In this case, it's the Standard Input Output Library, allowing us to make text appear on the screen (as well as other things).
- Almost every C program you will write in this course will have this line.

4 #include <stdio.h>
BREAKING IT DOWN INTO PARTS

THE "MAIN" FUNCTION (LINES 6-9)

- A function is a block of code that is a set of instructions
- Our computer will run this code line by line, executing our instructions
- The first line has details that we’ll cover in later lectures
  - `int` is the output type - this stands for integer, which is a whole number
  - `main` is the name of the function
  - `(void)` means that this function doesn’t take any input
BREAKING IT DOWN INTO PARTS

THE "MAIN" FUNCTION

- Between the { and } are a set of program instructions

```c
{}
```

- `printf()` makes text appear on the screen. It is actually another function from `stdio.h` which we included.

```c
printf("Hello\n");
```

- `return` is a C keyword that says we are now delivering the output of the function. A `main` that returns 0 is signifying a correct outcome of the program

```c
return 0;
```
In the Linux terminal we will open the file to edit
```
gedit helloWorld.c
```
Once we're happy with the code we've written, we'll compile it
```
dcc helloWorld.c -o hello
```
  - The `-o` part tells our compiler to write out a file called "hello" that we can then run
The `./` lets us run the program "hello" that is in our current directory
```
./hello
```
AND WE ARE OFF!

WE NOW HAVE OUR FIRST WORKING PROGRAM...

- Try this yourself!
- Try it using VLAB via your own computer
- Try setting up a programming environment on your own computer (differing levels of difficulty depending on your operating system)
Untold History of AI: Invisible Women Programmed America’s First Electronic Computer

The “human computers” who operated ENIAC have received little credit

IEEE Spectrum : Mar 25, 2019
<table>
<thead>
<tr>
<th>ADMIN</th>
<th>RESOURCES</th>
<th>HELP!</th>
<th>WHAT IS ...?</th>
<th>LINUX</th>
<th>C</th>
</tr>
</thead>
</table>
| How COMP1511 is run | Where to find resources (course webpage and forum) | How to get help and best ways to approach learning programming | What is programming? What is an Operating System? What is Linux? | Some basic Linux commands to get you started | Hello World!

**WHAT DID WE LEARN TODAY?**
Check out the forum

CONTENT RELATED QUESTIONS

Check out the forum

ADMIN QUESTIONS

cs1511@cse.unsw.edu.au