

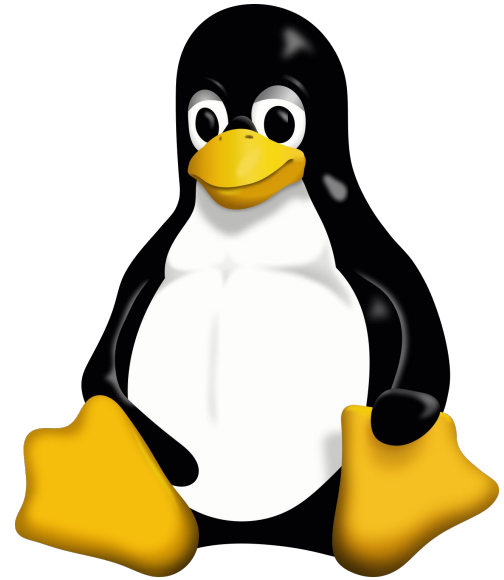
COMP1511/1911 Programming Fundamentals

Week 1 Lecture 1

Let's Start at the Very Beginning

Today's Lecture

- Welcomes and Introductions
- Getting help in COMP1511/1911
- COMP1511/COMP1911 admin details
- Working in Linux
- A first look into C



Ask and answer questions in the chat and/or put your hand up if you are here in person! 🙋💬

Who am I?



Dr Angela Finlayson

- Lecturer
- Oscar's mum
- Tina Arena Fan
- Civ 6 Enthusiast
- Kombucha Addict

Our Amazing Course Admins



Sofia De Bellis



Holly Field

Our Amazing Course Admins



Amber Lucas



Isabella Bywater

Our Wonderful Lecturer Moderators



**Liam
Phillips**



**Sophie
Moeskops**

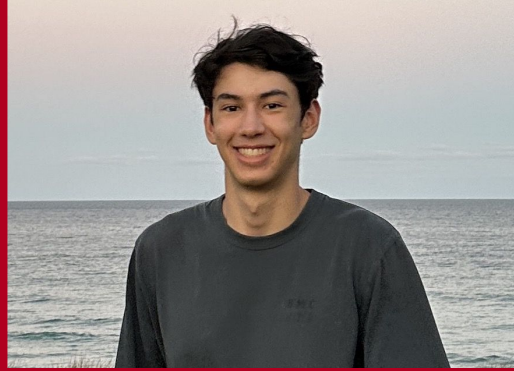


**Ethan
Scott**

Fabulous Forum Team



**Wesley
Donald**

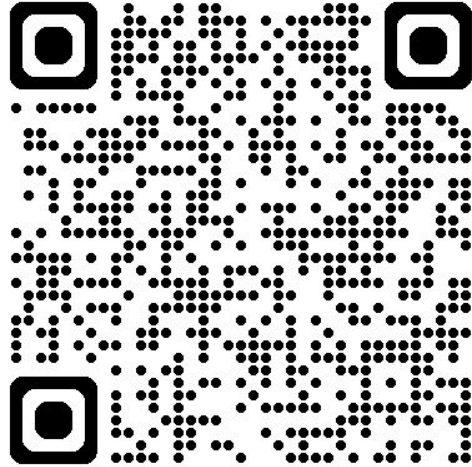


**Max
Davidson**



**Jackie
Maldon**

And our Incredible Team of Tutors!



[Meet the COMP1511/1911 26T3 Team!](#)

Getting to know COMP1511/COMP1911

We teach you the **fun**damentals of programming

- how to approach and **solve problems**
- how to express our solutions in step by step instructions that a computer can execute

We assume

- no prior knowledge
- no previous programming experience



Problem solving is a skill that can only be built with practice!

Code of Conduct

-  UNSW is committed to providing an **inclusive learning** environment
- All activity connected to COMP1511 and COMP1911, including social media, must be respectful 
 - No discrimination of any kind
 - No inappropriate behaviour
 - No harassment, bullying, aggression or sexual harassment
 - Full respect for the privacy of others

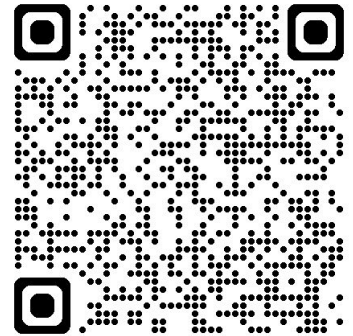
Getting Help In COMP1511/1911 and UNSW

Special Consideration

If illness or an emergency affects your study or assessment

- Apply for Special Consideration
- You will need supporting documentation
- You may be granted
 - Extension for assignments or labs
 - Supplementary exam

<https://student.unsw.edu.au/special-consideration>



Equitable Learning Plans (ELP)

For chronic conditions that affect your studies get an ELP

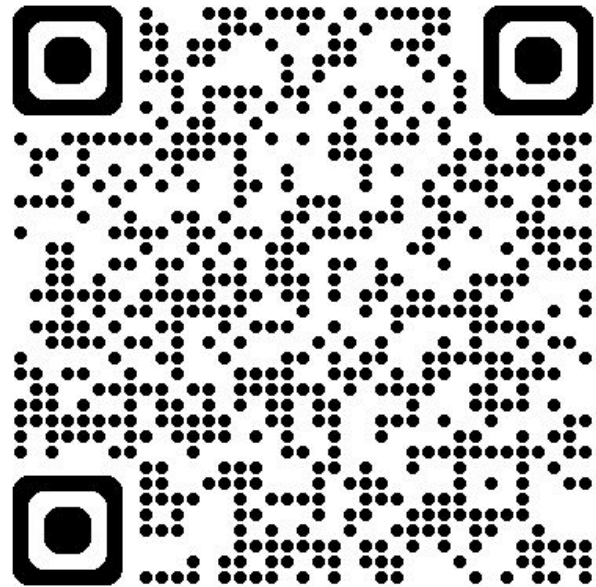
- Confidential
- Educational adjustments to provide support
- For more info, please visit:

<https://www.student.unsw.edu.au/equitable-learning>



Course Content Related Help

- Ask questions in **lectures** and in **lecture chat**
- Ask Questions in **tuts and labs!**
- **Forum**
 - Post all your questions here
 - Feel free to answer other's questions
 - Don't post your code publicly in the forum



<https://discourse01.cse.unsw.edu.au/26T3/COMP1511/>

Course Content Related Help

- **Help Sessions**

- One-on-one help outside your normal tut-lab
- No booking needed, start Week 3

- **Drop in Sessions**

- Longer one-on-one help
- Bookings needed, start Week 4

- **Revision Sessions**

- Group sessions to revise relevant topics
- Bookings needed, start Week 4

Schedules will be announced soon

Admin Related Help

- **Course Administration Issues:**

- Email: cs1511@unsw.edu.au

- **Enrollment Issues:**

- UNSW Nucleus Student Hub

<https://nucleus.unsw.edu.au/en/contact-us>

- **cse course account issues:**

- CSE Help Desk:

<http://www.cse.unsw.edu.au/~helpdesk/>

Who are you?

Say Hi to Someone!

In the room: Introduce yourself to someone sitting near you.

Online: Say hi in the chat 🙋

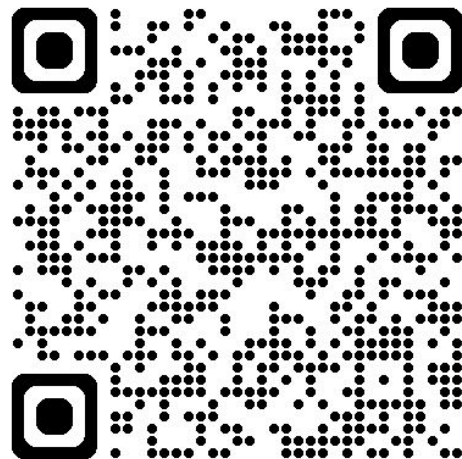
Getting to Know you Kahoot!

How COMP1511/1911 Works

Course Website

<https://cgi.cse.unsw.edu.au/~cs1511/current>

- All course information is on our course website
- Please bookmark it
- **We don't use Moodle**
- Please read the **course outline** thoroughly



Course Format

- Weekly Lectures 2 x 2 hours
- Weekly tut/labs 3 hour blocks
- 2 Major Assignments
- 1 Final Exam

Note: **Week 6** is Flexibility week. No lectures or tut/labs that week.

Lectures 2 x 2 Hours Each Week

Monday 11am-1pm

Ainsworth G03 + livestream

Tuesday 4pm-6pm

Livestream only

 **All lectures are live streamed on YouTube and recorded.**

Lectures

-  **We'll introduce concepts and work through coding examples**
-  **Ask questions:** put your hand up or use the live chat
-  **Slides, code and recordings:** available from the course website

Tutorial: 1 hour

- Interactive problem-solving with your tutor and class
- Tutorial questions are available in advance
- Online or face-to-face, check your timetable for your class
- For online classes:
 - Use Teams
 - Turn on your cameras if you can
 - Pets appearing on camera are very welcome 🐕🐈🦜🐰🐢

Lab: 2 hour (directly after tutorial)

- **Labs are where most of the learning happens**
 - Hands on practical coding
 - Support and feedback from tutors and lab assistants
- **Each week includes**
 - **Check off** activity
 - **Problem set** exercises
- **To get marks for the lab**
 - **you MUST attend your Check off in class time** 🚓
 - you can then get marks for the problem set exercises you submit

Lab Check-off: Every Week in Your Lab

 **To be eligible for that week's lab marks you MUST:**

- Attend your lab
- Complete a check-off with your tutor

 **What happens in a check-off?**

- Your tutor will show you some code and ask you a few questions
- You pass by making an attempt, not being perfect
- You will get feedback to help you learn and guide your revision!

Week 8 and **Week 10**: The check-off will be practice exam questions

Lab Check-off: Every Week in Your Lab



What if you are sick and can't attend

- If you cannot make your scheduled tut-lab for a certain week you can **attend any other tut-lab that week** and get checked off there.
- If you cannot attend any tut-lab at all that week, please apply for **Special Consideration**.
- Email cs1511@unsw.edu.au if you have any questions

Lab: Problem Sets

- **Purpose**
 - Hands on coding practice.
 - Mostly individual, some pair programming exercises
- **Released:** each **Sunday 6pm**
- **Due:** **Monday 6pm** the next week
- **Marks:** 15% overall
 - 2 marks each week capped at 15 (no marks for week 1)
- **Difficulty levels**
 - 1 dot and 2 dot standard exercises
 - 3 dot optional challenge exercises for bonus marks

Assignments

Assignment 1

20%

Released Week 4 Due Week 7

Assignment 2

25%

Released Week 7 Due Week 10

- Individual work
- Designed to take a few weeks
- Apply the theory and practice your skills on larger problems

Late Penalties

- Assignments and Labs can be submitted late with a penalty
- Late penalties of 5% per day late apply off the ceiling
 - maximum lateness is five days, after which time it is zero marks

Plagiarism

- **Plagiarism:** The presentation of someone else's work or ideas as if they were your own.
- **For individual assessments, your submission must be entirely your own work.**
 - Collaboration on individual assessments is plagiarism
- Any kind of cheating on your work for this course will incur penalties (see the course outline for more details)
 - Potentially lose the marks for the assignment
 - Or be asked to leave UNSW
- **We have sophisticated plagiarism detection software**

Use of AI Tools

- You are not permitted to submit code generated by automatic tools such as ChatGPT, Claude, GitHub Copilot, Google Gemini
- This is treated the same as plagiarism

Quick question: Why else is it a bad idea to use AI to write your code for this course?

Dangers of Over-Relying on AI

Programming a skill you learn by practising!

- Using AI means you miss opportunities to practise
 - Problem Solving
 - Structuring your code and data
 - Testing, Debugging
 - Get satisfaction of getting code you wrote yourself to work!!!
- There is no AI in the final exam
- You need to understand the foundations for future courses

Catching ubers won't help you train for a marathon!

Final Exam: 3 hours

- **In Person**

- 3 hours
- C programming questions

- **Exam Hurdles**

- Parts of the exam are competency hurdles
- Hurdles must be completed satisfactorily to pass the course
- Array Hurdle for COMP1511 and COMP1911
- Linked List Hurdle for COMP1511 but NOT COMP1911
(COMP1911 students still do linked list questions for marks).

Supplementary Exam

- May be offered to students granted Special Consideration
- Identical in format to the main exam
- Held in January/February 2027 (exact date pending)
 - you must make yourself available if you are granted a supplementary exam
- [Fit-to-Sit rule](#)

Total Assessment

Labs = 15%

Assignment 1 = 20%

Assignment 2 = 25%

Final Exam = 40%

To pass the course you must:

- Score at least 50/100 overall in the course
- Pass the exam hurdle/s

COMP1911 vs COMP1511

- **The courses are almost identical**
- **COMP1911**
 - Designed for non-computing majors
 - A prerequisite for COMP1521, but not for other COMP courses
 - Does not require some final stages of Assignment 2
 - Does not have a Linked List exam hurdle
 - COMP1911 students still answer Linked List questions for marks.

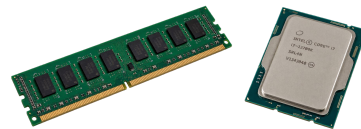
Quick Break. Back Soon...



Programming in C on Linux

Computers and Programs

- Computers can do incredibly **complicated things...**
but everything they do is made up of **simple instructions**
- **A program is a set of instructions for the computer to follow.**
- To run them the computer has:
 - **Memory:** stores instructions and data
 - **Processor:** executes instructions



Programming

- Writing **instructions** (code) to solve a problem
 - Programming involves a cycle of:
 - think
 - code
 - test
 - debug
- You may go through many iterations before you get it right
- Don't skip the think stage:
 - You need to understand the problem you are trying to solve
 - You might need to start with pen and paper planning first

Mistakes and Bugs are Normal

At times you may require

- patience
- perseverance

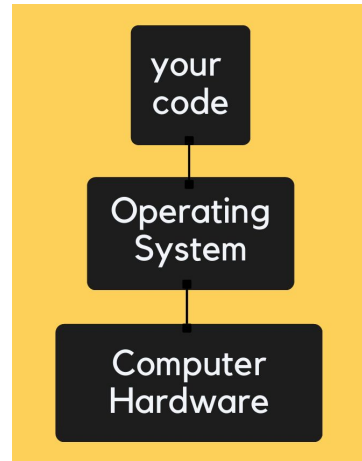
The key to success is

- practice



Operating Systems

- An Operating System is the interface between the user and the computer hardware
 - Execute user programs
 - Make sure programs do what they're supposed to
 - Schedules access to limited resources (hardware)
 - Make the computer system convenient to use
- Operating systems are themselves programs!
 - Which one are you using?



In this course we will use Linux because

- Consistency and fairness
 - everyone uses the same environment
- Industry relevance
 - widely used in servers, research and development
- Transparency
 - open-source tools show how systems really work
- We love it. We hope you love it too!



We teach you to look like a hacking genius!



Do I need to buy a Linux Computer?

- No CSE Provides the Linux Environment you Need
- On Campus:
 - CSE Labs have everything all set up
- At Home:
 - Use **VLAB** to remotely access CSE's linux computers
 - You can get help setting this up in your first lab

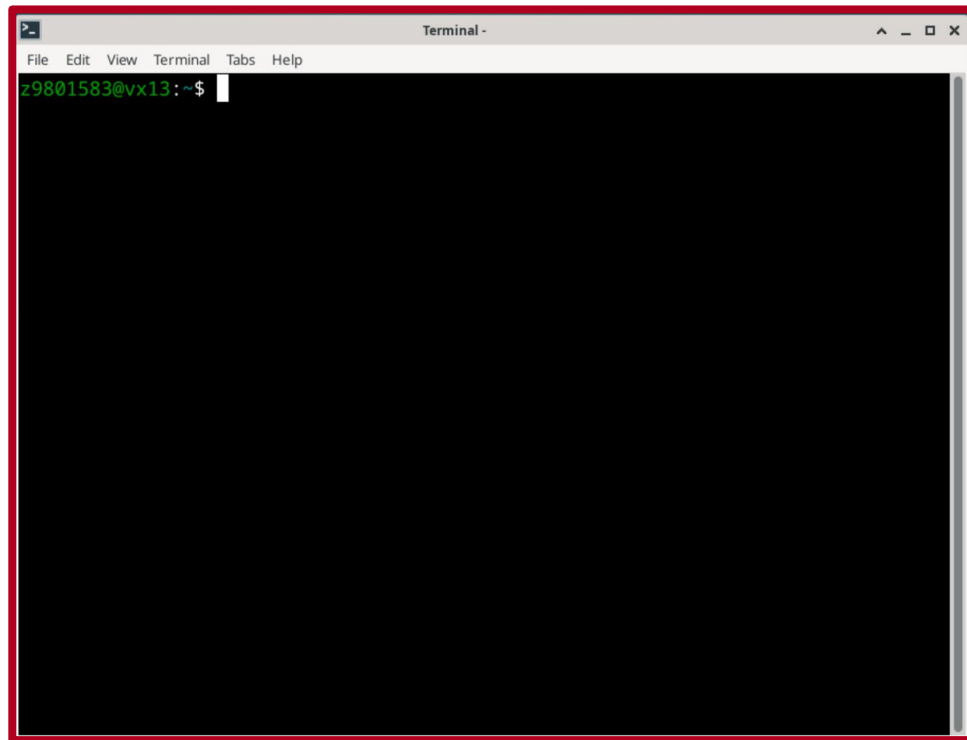
https://cgi.cse.unsw.edu.au/~cs1511/current/resources/home_computing.html

Working in your CSE account

- Log into **VLAB** or CSE lab machine
 - with zid and zpass
- Open a terminal
- Type **1511 setup** in the terminal to get everything ready
 - you will do this in your first Lab.
 - you only need to do it once

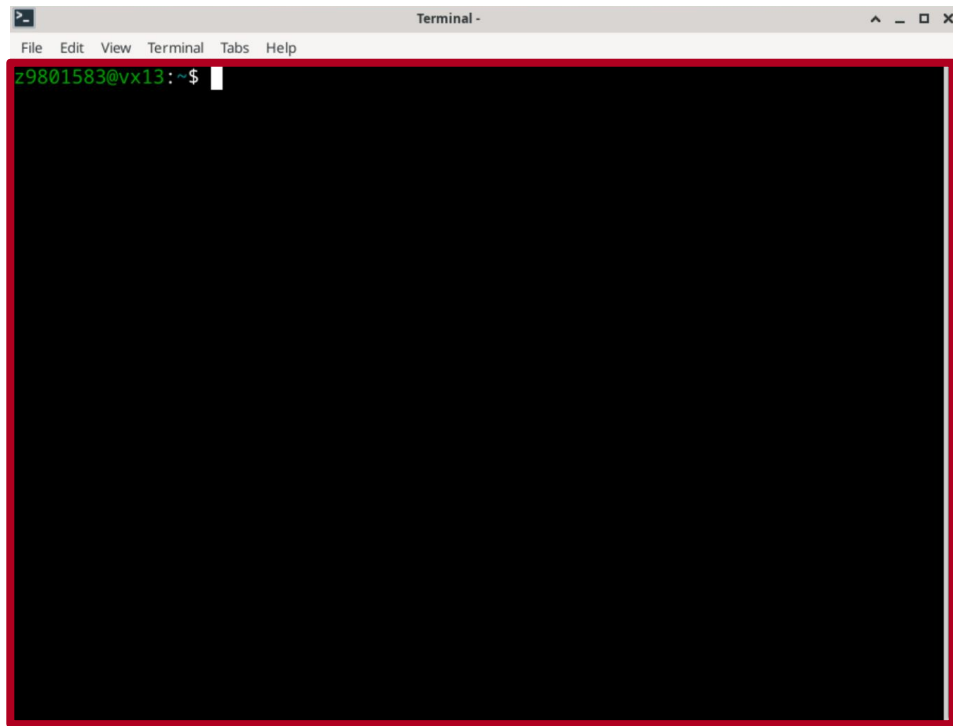
The Terminal

- A graphical application
 - sends text commands to the shell
 - displays shell output



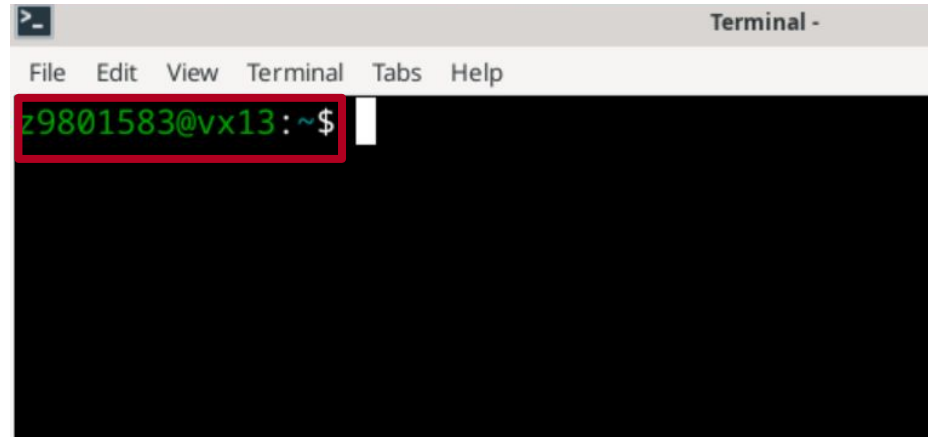
The Shell

- The shell, (bash, zsh) is a program that runs in the terminal.
- It executes commands, and has its own syntax.
- It returns output which the terminal can display



The Command Line Prompt

- The prompt is controlled by the shell
- It is the line of text which displays some information often ending in a \$
- Lets you know it is ready for you to type

A screenshot of a terminal window titled "Terminal -". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal content shows a green prompt string "z9801583@vx13:~\$" followed by a white cursor. A red rectangular box highlights the prompt string.

Important Linux Shell Commands

<code>ls</code>	List all the files in the current directory
<code>mkdir dir_name</code>	Make a new directory called dir_name
<code>cd dir_name</code>	Changes the current directory to dir_name
<code>cd ..</code>	Move up one directory (folder) level
<code>cd</code>	Move to your home directory
<code>pwd</code>	Tells you where you are in the directory structure

Important Linux Shell Commands

<code>cp source destination</code>	Copy a file from the source to the destination
<code>mv source destination</code>	Move a file from source to destination directory OR Rename source to destination
<code>rm file_name</code>	Remove (delete) file_name Does not put in trash. Permanently deletes Warning: Use <code>rm</code> with caution

Important Linux Shell Commands

The `-r` flag can be added to `cp` or `rm` commands to recursively go through a directory and perform the command on all the files

`cp -r source destination`

copy all files from source to destination directory

Find out more tips and tricks in the [Linux Cheatsheet](#)

Programming Languages

- **Humans**

- Humans speak in languages that are often ambiguous and not precise enough for computers.

- **Computers**

- Understand binary: 0s and 1s
 - 0100 0000 0000 0000 0000 0000 0000 0000
- Instructions written in binary are precise but hard for humans to read, write and reason about.

Programming Languages

- **Computer Programming Languages**
 - A compromise
 - Easier for humans to understand than binary!
 - Precise enough to be translated into binary
- **Computers do not understand Programming Languages**
 - We need another program to map our programming languages into 0s and 1s our computer can understand.

Why Program in C?

- **C helps us understand how data is stored in memory**
- **C is widely used** in operating systems and embedded systems
- Many programming languages were influenced by C
- We learn C to
 - Learn fundamental programming and problem solving skills
 - Build skills that will transfer to other programming languages



Learning C makes other programming languages easier to learn and helps you understand what is going on behind the scenes.

Finally... we get to C some C

```
// A program showing how to print output in C
// The first of many C programs you will C

#include <stdio.h>

int main(void) {
    printf("Hello COMP1511 and COMP1911\n");
    return 0;
}
```

Let's Break it Down: Header Comment

- `//` in front of a line makes it a comment
- If we use `/*` and `*/` everything between them will be comments
- Comments are for humans
- The compiler ignores them
- They help future selves and colleagues understand our code
- Header comment is a special comment we put at the top of each program

```
// A program showing how to print output in C
// The first of many C programs you will C
```

```
#include <stdio.h>
```

```
int main(void) {
    printf("Hello COMP1511 and COMP1911\n");
    return 0;
}
```

Let's Break it Down: #include

- `#include` lets the compiler know what part of the standard C library we want to use
- In this case, it's the Standard Input Output Library
 - we need it for `printf`
- Almost every C program you will write will have this line

```
// A program showing how to print output in C  
// The first of many C programs you will C
```

```
#include <stdio.h>
```

```
int main(void) {  
    printf("Hello COMP1511 and COMP1911\n");  
    return 0;  
}
```

Let's Break it Down: main function

- Every C program needs exactly 1 main function
- That is where program execution starts
- Between the { and } are a set of program instructions
- Our computer will execute the instructions line by line
- A main that returns **0** signifies that the program executed successfully.

```
// A program showing how to print output in C  
// The first of many C programs you will C
```

```
#include <stdio.h>
```

```
int main(void) {  
    printf("Hello COMP1511 and COMP1911\n");  
    return 0;  
}
```

Let's Break it Down: printf

- `printf()` writes text to standard output (screen).
- It is a function from `stdio.h` which we included
- Everything between the starting "`"` and ending "`"` gets printed
- `\n` is the newline character
- All executable statements end with a semicolon `;`

```
// A program showing how to print output in C
// The first of many C programs you will C

#include <stdio.h>

int main(void) {
    printf("Hello COMP1511 and COMP1911\n");
    return 0;
}
```

Demo: Writing your first C program

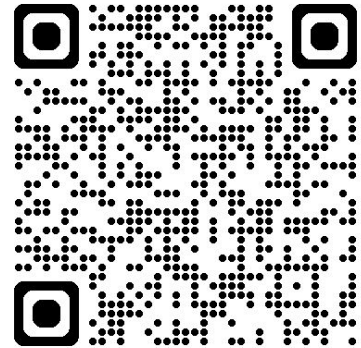
- Follow along now or try it later once you have set up VLAB
- Open a file called `hello_world.c` in VSCode from the terminal by typing

```
code hello_world.c
```

- Edit the file and save it

- You can find live lecture code at

https://cgi.cse.unsw.edu.au/~cs1511/26T3/code/week_1/



But how can we actually run our program???

- Computers can't understand or execute C code
- Compilers are programs that turn code into machine code

We use the **dcc compiler**

- **dcc** was developed at CSE UNSW
 - Designed for beginners and gives helpful error messages

Demo: Compiling and running code

- Compile a C program `hello_world.c` into an executable file called `a.out`

```
$ gcc hello_world.c
```

- Run our executable file:

```
$ ./a.out
```

Note: `./` just means the current directory

Demo: Compiling and running code

What if we have lots of executable programs and we want to give them names other than a.out?

Compiles a C program hello_world.c into an executable file called hello_world

```
$ gcc hello_world.c -o hello_world
```

Run our executable:

```
$ ./hello_world
```

Compile errors or warnings?

1. Try to fix the first error first
2. Save
3. Re-compile
4. Repeat till errors are gone

Then you can run your program to test that it does what you want!
You may need to edit it, compile and repeat...

Compile errors or warnings?

Error on line 42

```
41      });  
42  
43      if (includ
```



dcc-help and dcc-sidekick

`dcc-help` and `dcc-sidekick` are tools to help beginners understand compiler error messages to help you fix your code.

Great to use while learning and doing labs and assignments

Warning: `dcc-help` and `dcc-sidekick` are not available in the final exam. Note that you WILL still have `dcc` in the final exam.

Escaping

- \ is an escape character
 - \n is an escape sequence that means newline
 - \t is an escape sequence that means tab
 - \" is an escape sequence that allows us to print the " character
 - Why would we need this?

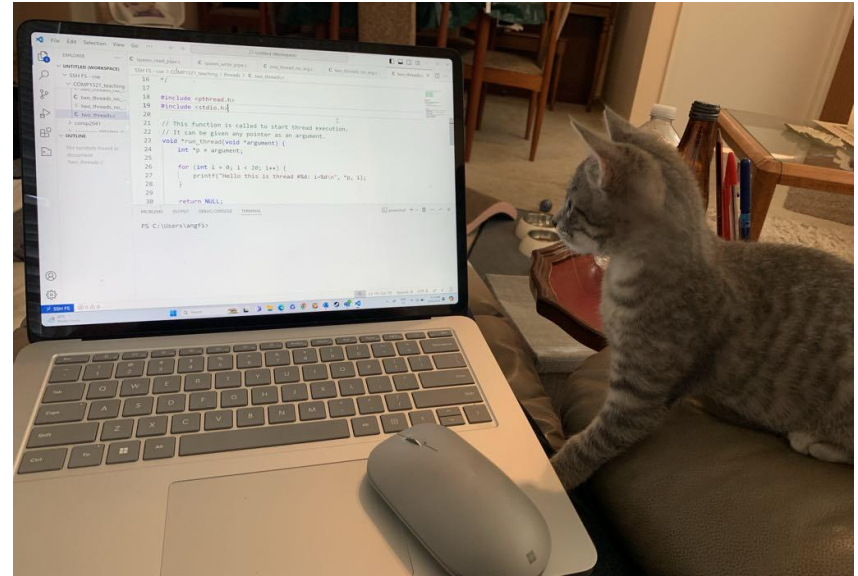
How can we actually print a \ character?

Why we use \\ of course!

```
printf("This is a Backslash \\");  
printf("This is a Backslash followed by a newline \\n");
```

Remember that Learning Programming

- Requires practice
 - Labs and assignments are a great way to learn!
 - Try things and make many many many mistakes!
 - Ask for help when needed
- It can be challenging but also very rewarding!
- We are here to guide you and support you.



What did we learn today?

- **Admin:** How the course is run
- **Resources:** Where to find them
- **Help:** Where to find it
- **Concepts:** Computer, Program, Operating System
- **Linux:** Some basic commands
- **C:** Compiling and running your first C program
 - `hello_world.c`

What will we learn tomorrow

- Variables
 - How can we **store data** in our programs
- How can we **read in data** from users
- Expressions
 - How can we do **calculations** on our data

Reach Out

Content Related Questions:

[Course Forum](#)

Admin related Questions email:

cs1511@unsw.edu.au



Student Support - I Need Help With...

My Feelings and Mental Health

Feeling depressed, overwhelmed
or not your usual self?



Mental Health Connect
student.unsw.edu.au/mhc



TalkCampus
student.unsw.edu.au/Talkcampus



UNSW (24/7) Mental Health Support: [+61\(2\) 9385 5418](tel:+61293855418)

Text support After Hours: [0485 826 595](tel:0485826595)



Offshore
Call 24-hour Medibank Hotline: [+61 2 8905 0307](tel:+61289050307)

Uni and Life in Australia

Stress, financial, visas,
Accommodation & More



Student Support
Indigenous Student Support

student.unsw.edu.au/advisors
nura-gili-centre-indigenous-programs

Reporting Sexual Assault/Harassment



Equity Diversity & Inclusion (EDI)

edi.unsw.edu.au/sexual-misconduct

Educational Adjustments

To manage my studies and
disability / health condition



Equitable Learning Services (ELS)

student.unsw.edu.au/els

Academic and Study Skills



Academic Skills

student.unsw.edu.au/skills

Special Consideration

Because life impacts our
studies and exams



Special Consideration

student.unsw.edu.au/special-consideration

Physical Health

Doctors visits, Vaccinations



Health Service

student.unsw.edu.au/hsu