COMP1511/1911 Programming Fundamentals

Lecture 1

Today

- Welcome and introduction
- Course administration
- How COMP1511 works
- How to get help
- The best ways to learn programming
- What is programming?
- What is Linux?
- Working with Linux
- A first look at C

Who am I?

- Dr Henry Hickman
  - PhD Computer Science Education
    - Automated assessment of introductory programming
  - Stand-up comedian
  - Loves Survivor a normal amount
  - Yell at me if I write Python
    - Chocolate for calling me out

The Admin Team

- Sofia De Bellis
  - Graduated from Computer Science @ UNSW
  - Software Engineer @ Atlassian
  - Love travelling, most recently Japan, Hong Kong and Taiwan
  - idk last fact for you to decide Henry, feel free to steal from Sasha idk what she put there ahahaha

The Admin Team

- Holly Fields
  - Volleyball legend
  - Took 3 years of a CS degree before buying a keyboard
  - Needs glasses (probably)

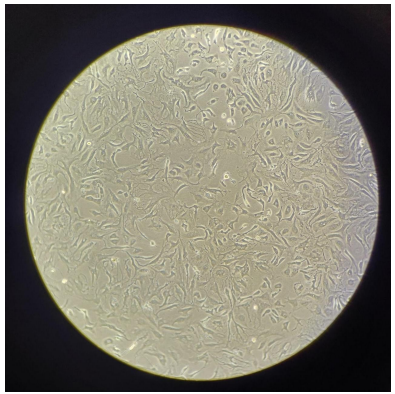
The Admin Team

- Grace Murray
  - Studied mechatronics @ UNSW
  - Can’t beat Bella’s rubix cube time...yet!
  - Can do the worm
    - Wil show us if myExperience response rate is high

The Lecture Mods

- Isabella Bywater

The Lecture Mods

- Isabella Bywater
  - 

The Lecture Mods

- Sophie Moeskops

The Lecture Mods

- Sophie Moeskops
  - 

The Lecture Mods

- Liam Phillips

The Lecture Mods

- Liam Phillips
  - 

The Wonderful Tutoring Team!

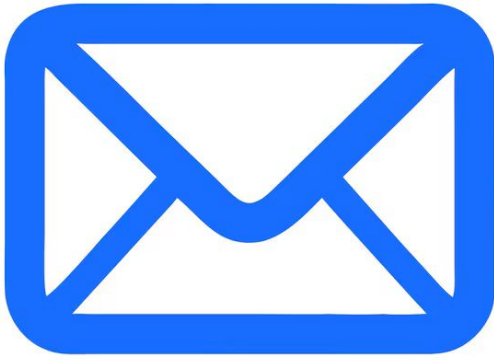
- https://cgi.cse.unsw.edu.au/~cs1511/26T2/team/

Who are You?

- Introduce yourself to your neighbour!
  - This will help you learn to program I swear

Communicating with the Course - Admin

- You may want to email me
  - Don't
- All admin related enquiries go to:
  - cs1511@unsw.edu.au
- All enrollment issues
  - UNSW Nucleus Student Hub
  - https://nucleus.unsw.edu.au/en/contact-us
- ELP Plans
  - Make sure they are in place
  - https://www.student.unsw.edu.au/equitable-learning



Communicating with the Course - Getting Help

- Ask questions in lectures
- Ask questions in tuts/labs
- Post all your questions on the course forum
  - https://cgi.cse.unsw.edu.au/~cs1511/26T2/resources/forum.html
- Help sessions will exist

What is COMP1511

- Introduction to Programming
- Computers follow instructions that we give them
  - Writing a program == telling a computer to do something
- More than anything, this is a course on problem solving
  - Practice makes perfect

COMP1511 vs COMP1911

- Not actually a fight
- All content taught is the same
  - COMP1911 will have slightly reduced assessment requirements
    - No linked list hurdle (we'll explain that soon)
    - Slightly smaller Assignment 2 (we'll also explain that soon)
- If you aren't sure which course you're enrolled in, email cs1511@unsw.edu.au

Course Format

- We assume you know nothing
  - No programming experience needed
- We teach you:
  - the fundamentals of programming
  - how to approach and solve problems
  - how to talk to computers in a common language

Lectures

- Monday 11am-1pm
  - J17 G03 & Online via YouTube
- Tuesday 4pm-6pm
  - Online via YouTube
- Recordings available
- Week 6 is flex week (no lectures)
- If you have questions, leave them in chat
- Please be respectful of others
  - There are more students than chairs, if you don’t want to be here, someone else does

IMPORTANT NOTE ABOUT WEEK 2 DO NOT SKIP PLEASE

- WEEK 2 MONDAY IS A PUBLIC HOLIDAY
  - THERE WILL BE NO IN PERSON LECTURE ON WEEK 2 MONDAY
- HENRY WILL POST AN EPIC YOUTUBE VIDEO FOR YOU ALL INSTEAD
  - HENRY IS SORRY, BUT THIS IS IMPORTANT FOUNDATIONAL CONTENT THAT YOU NEED TO LEARN
- THERE WILL BE NO TUTORIALS ON MONDAY
  - WE WILL SEND OUT AN ANNOUNCEMENT WITH A LINK TO MAKEUP TUT LABS FOR NEXT WEEK ONLY

Lecture Content

- Why bother?
  - Why what we're learning is important
- Theory
  - What we are trying to understand
- Demonstrations
  - Lots of live coding to show off
- Problem solving
  - How do we decide what to code?
- Other stuff
  - Outside of programming, what's important?

Lecture Resources

- Slides can be found here:
  - https://cgi.cse.unsw.edu.au/~cs1511/26T2/
- Lecture recordings will be on YouTube and linked on the same website

Tutorials

- One hour class environment
- Go further in-depth on the stuff we cover here
- Actual practical working tasks
- Learning to solve problems before you write code
- Questions available in advanced online
  - https://cgi.cse.unsw.edu.au/~cs1511/26T2/
  - Read these beforehand

Tutorials

- Online and face-to-face
  - Please check your timetable
- Online is via Teams
  - Please turn on your camera if you can
  - Please show us cool pets if you can
- Sample answers released after last tutorial for the week

Labs

- Practical coding involving small working groups
- Time to have one-on-one conversations with your tutors
- Problem sets will be marked automatically and count towards final grade
  - 15%
- There are challenge exercises for bonus marks
  - NOT NECESSARY
  - VERY CHALLENGING
- No tutorials or labs in Week 6

Check Offs

- Each week you need to attend your lab and complete a check-off with the tutor.
  - You are required to do the check-off to receive marks from that weeks labs.
  - The check-off consists of your tutor showing you some code and asking you a few questions about it.
- 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, please apply for Special Considerations.
- Any questions, please email cs1511@unsw.edu.au!

Assignments

- Larger scale projects
- Individual work
- These will take a few weeks and will test how well you can apply the theory you've learnt

Assignments

- There are two of them
  - Assignment 1 - 20%
    - Due Monday 6pm Week 7
  - Assignment 2 - 25%
    - Due Friday 6pm Week 10
- Late penalties of 5% per day apply off the ceiling
  - After 5 days you get 0 marks

Help Sessions

- Optional
- Run in person and on Teams
- Face-to-face help sessions will have allocated lab space
- Some one-on-one consultation with tutors
- Time to ask individual questions or get help with specific problems
- Schedule will be on the course website soon
- These are super busy around Assignment deadlines

Final Exam

- INVIGILATED IN-PERSON LAB EXAM
- 3 hours total
- You will be given a series of problems to solve in C
- You will be expected to read C and show you understand it
- There will 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
  - Problems using arrays
  - Problems using linked lists

Total Assessment

Labs	15%
Assignment 1	20%
Assignment 2	25%
Exam	40%

- To pass the course you must
  - Score 50 out of 100
  - Solve problems using arrays in the final exam
  - Solve problems using linked lists in the final exam

Special Consideration

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

Equitable Learning Plans

- Please make sure your ELPs are sorted so we can provide you the best support possible
- For more information:
  - https://www.student.unsw.edu.au/equitable-learning

Supplementary Assessment

- A supplementary exam can be offered to students granted special consideration
- Identical format to the main exam
  - Different questions
- Held some time later
  - If you have a supplementary assessment, please stay in Sydney
- Fit to Sit rule
  - Please take this seriously

Code of Conduct

- All students have a right to learn
  - 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

Plagiarism

- Presenting someone else's work or ideas as your own
- Any kind of cheating will incur penalties
  - See course outline for details
- Collaboration on individual work (like Assignments) is considered plagiarism

Collaboration vs Plagiarism

- Discussion of work and algorithms is fine
  - It’s encouraged, even
- The internet has a lot of resources you should use
  - Just cite your sources
- No collaboration at all on individual assignments
- Your submissions are your own work
- Do not use other people’s code
- Do not ask other people to solve problems for you
  - Even verbally
- Do not provide your code to other people
  - This will get you both in trouble

Collaboration vs Plagiarism

- At best you'll lose some marks
- At worst you'll be asked to leave UNSW

Use of AI

- Using AI to learn to code is like using a scooter to train for a marathon
  - The struggle is the point
  - Learning is hard
  - What are you actually trying to accomplish here?

Why You Shouldn’t Use AI

- There is no AI in the exam
  - If you have used AI during the term, you will likely crash out in the exam
    - If you crash out in the exam, you will fail the course
- Learning to program is a rewarding experience
  - Don’t rob yourself of the fun because you want an extra fraction of a percent of a mark

Why You Shouldn’t Use AI

- Using AI will get you to the right answer, but you miss the critical learning phase, and stunt your own development
- Programming is hours and hours of repetition
  - You are learning a new language
  - It is rewarding and fun and wonderful and frustrating at times
    - But it’s so worth it!

If You Want More Info

- Course webpage
- Course forum
- Recorded lectures
- One on One
  - Ask your tutor
  - Go to help sessions

Serious Issues

- Email
  - cs1511@unsw.edu.au
- The Nucleus
  - nucleus.unsw.edu.au
- CSE Help Desk
  - https://www.unsw.edu.au/engineering/our-schools/computer-science-and-engineering/student-life/cse-it-helpdesk

Student Support | I Need Help With...

My Feelings and Mental Health
Managing Low Mood, Unusual Feelings & Depression

Mental Health Connect — student.unsw.edu.au/**counselling** — Telehealth

Mind HUB — student.unsw.edu.au/**mind-hub** — Online Self-Help Resources

In Australia Call Afterhours
UNSW Mental Health Support Line — 1300 787 026 — 5pm-9am

Outside Australia Afterhours
24-hour Medibank Hotline — +61 (2) 8905 0307

Uni and Life Pressures
Stress, Financial, Visas, Accommodation & More

Student Support — student.unsw.edu.au/**advisors**
Indigenous Student Support — nura-gili-centre-indigenous-programs

Reporting Sexual Assault/Harassment

Equity Diversity and 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**

Learning is Hard

- Secondary skills (like programming) take time to learn
  - Slow, deliberate, conscious effort
- It will not happen overnight
  - It will take time, effort, and repetition
- Do not feel disheartened if you don’t understand
  - It is part of the learning process
  - Try again and again and again
- Let us know if we can help
  - Reach out to tutors
- Attempt all your lab questions and assignments
  - Working through problems teaches you how to solve similar problems
- You can do it :)

Break Time

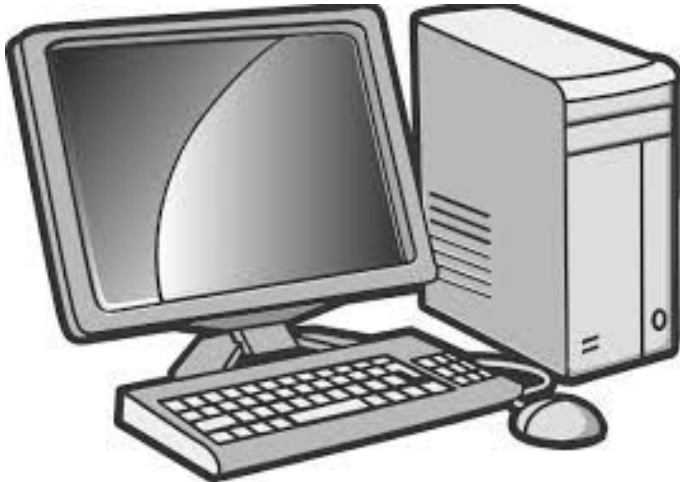
...probably...

Why Bother?

- Computers are extremely powerful systems
  - and yet everything they do comes from relatively simply commands
- We want to talk to the computer
  - Tell it to perform tasks for us

What is a Computer?

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



What is Programming?

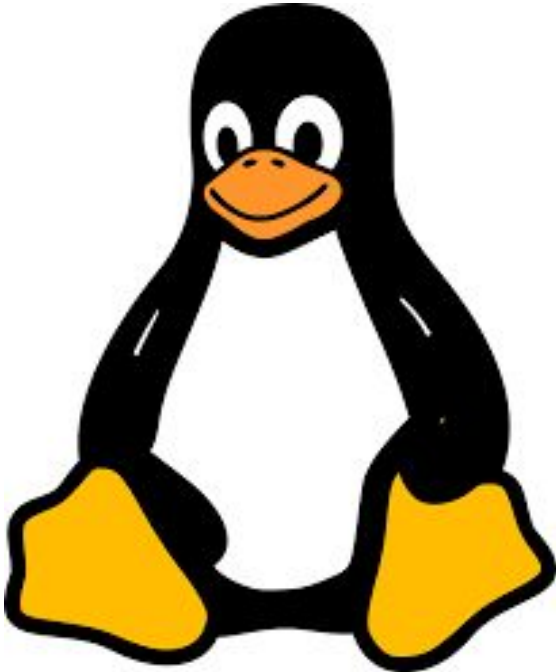
- Providing a computer with specific instructions to solve various problems
  - Use specific language to write those instructions
    - Code
- Problem solving
  - Mistakes are good
  - It may take many attempts to get the answer right

What is an Operating System?

- The interface between the user and computer
- Operating systems
  - Execute user programs and makes problem solving easier
  - Makes the computer system convenient to use
- Operating systems sit between our code and the computer, providing essential service

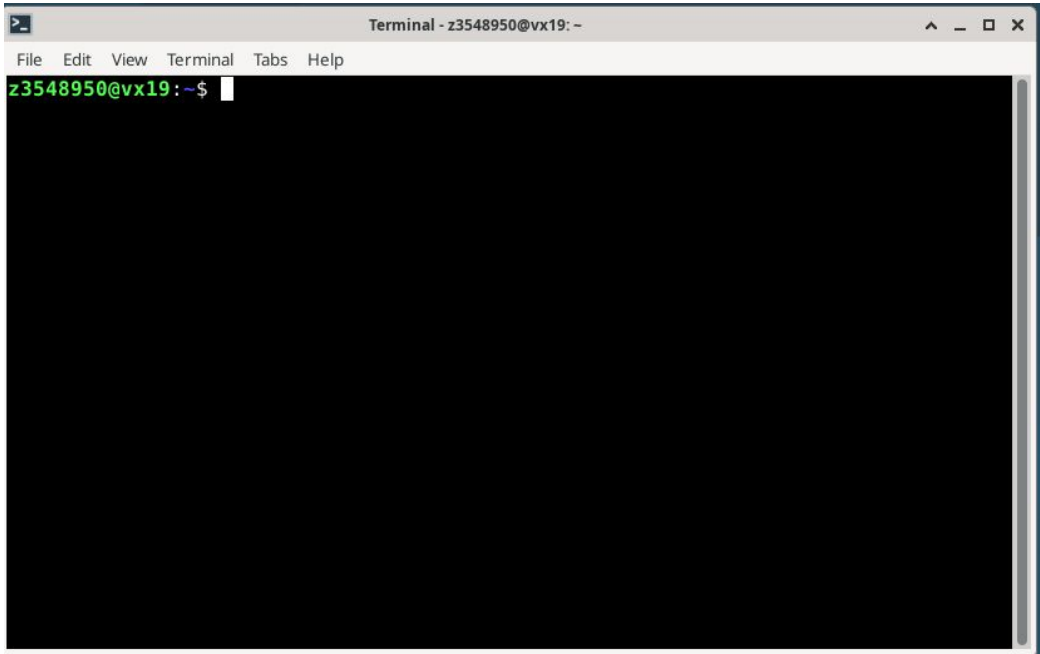
What is Linux?

- Unix based operating system
  - Open source
  - More reliable
  - Lightweight
  - Faster
  - More secure
  - Plays nicely with C



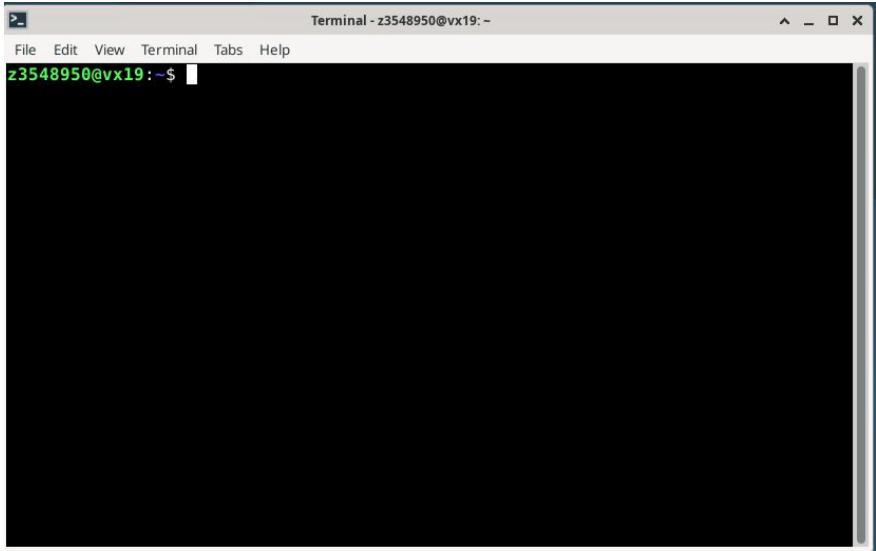
Terminal

- Allows us to send simple commands to the shell
- Handles things like
  - User input
  - Displaying output



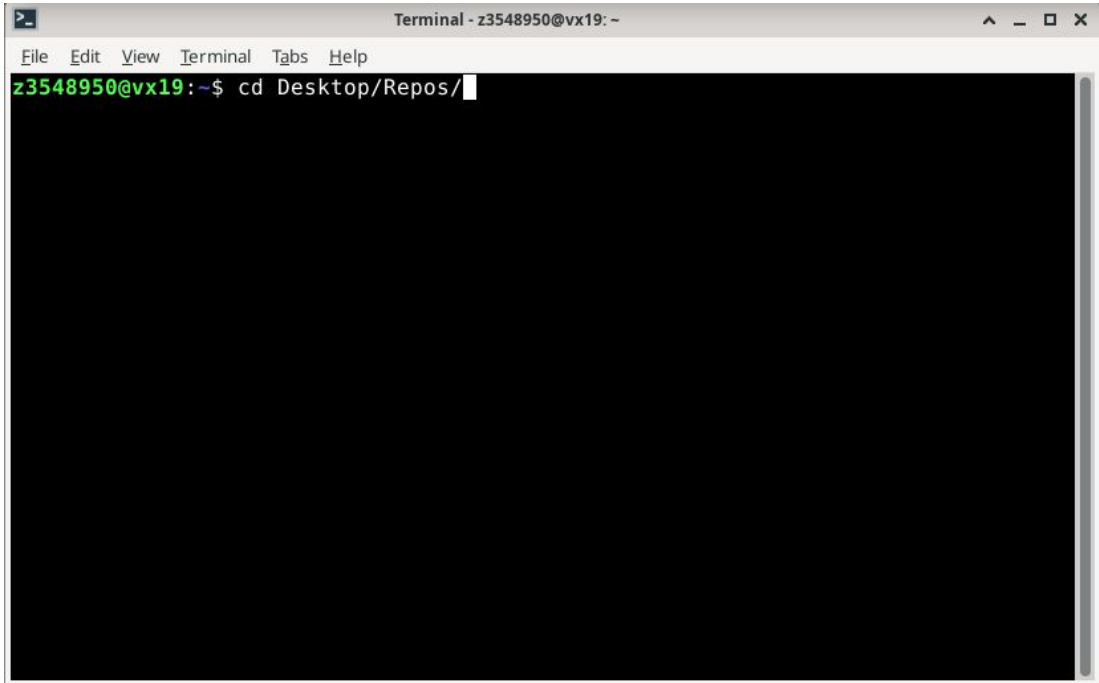
Shell

- Primary interface for the computer
  - A program that executes commands
  - Has its own syntax
  - Returns output
    - Terminal displays that output
  - Can launch other applications



Prompt

- The text we type into the shell



Important Terminal Commands

- ls
  - Lists all current files in directory
- mkdir directoryname
  - Makes a new folder called directoryname
- cd directoryname
  - Changes current directory to directoryname
- cd ..
  - Moves us up one level in our working directory
- pwd
  - Prints the working directory
    - Where we currently are

Command Line File Operations

- cp source destination
  - Copies a file from source to destination
- mv source destination
  - Moves a file from source to destination
- rm filename
  - Deletes a file
- cp -r COMP1511 COMP1511_backup
  - Will copy all files in COMP1511 to COMP1511_backup
  - -r can be added to cp or rm to do it for the entire folder
    - -r means recursively

Using CSE's Computing Resources

- Our labs run Linux with the tools needed to get started
- You will want your own computer ready to code
  - 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
    - Check the Home Computing site or the guides on our course website

For COMP1511 We Need

- A development environment
  - We use a minimal version of vscode
    - Run 1511 setup to get everything ready
      - You'll do this in the first lab
- A compiler
  - We use dcc
  - Takes our formal, human readable, C code and turns it into a machine readable program
    - The result is a program we can run
- You can use VLAB to access CSE editor and compiler

Programming in C

- We need a shared language to have a conversation
  - We have chosen C
- C is
  - A clear language with defined rules
    - Nothing we write will be ambiguous
  - A good starting point to control a computer from its roots
- Many modern programming languages are based on C

Our First C Program

```c
// A demo program
// Welcome to COMP1511 :)

#include <stdio.h>

int main(void) {
    printf("Welcome to COMP1511\n");
    return 0;
}

```

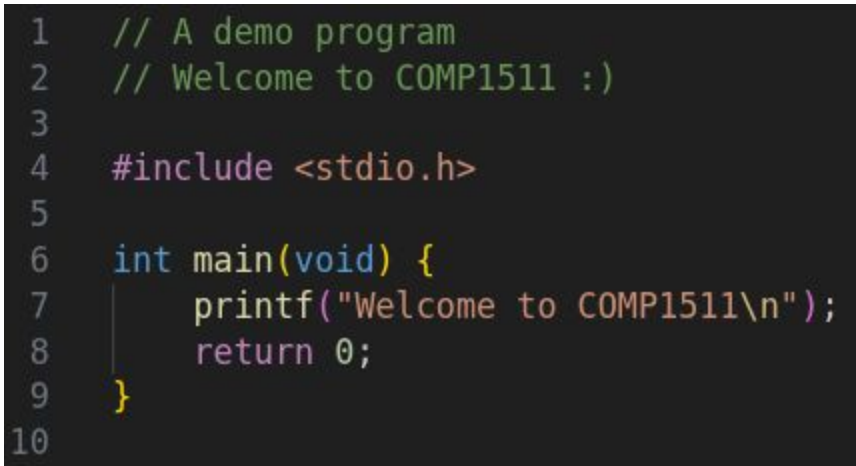
Breaking It Down

- Lines 1 & 2 are comments
  - Words for humans
- // in front of a line makes it a comment
- /* */ makes everything between them comments
- The compiler ignores comments
- Comments are important
  - We write programs for people

```c
// A demo program
// Welcome to COMP1511 :)

#include <stdio.h>

int main(void) {
    printf("Welcome to COMP1511\n");
    return 0;
}

```

Breaking It Down

- Line 4 asks the compiler to grab another file and add it to our code
  - In this case, it's the Standard Input Output library
    - Lets us put text on screen
- Almost every C program we write will include this



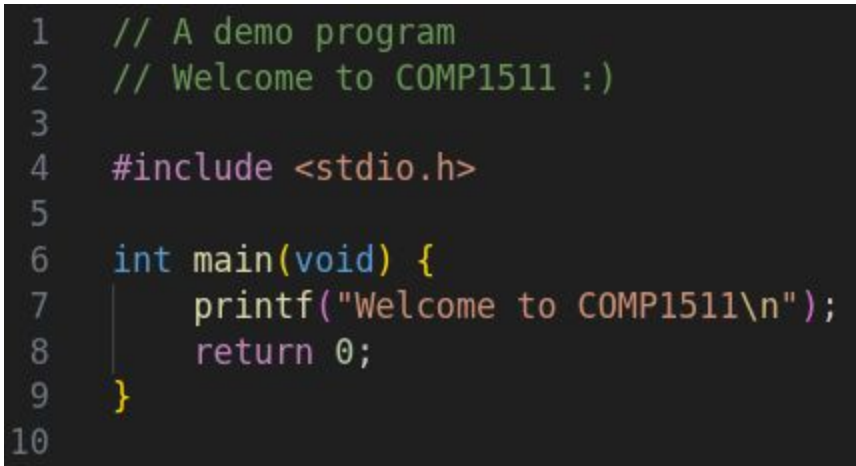
Breaking It Down

- Lines 6-9 are our main function
  - A function is a block of code that sets instructions and returns something
- Our computer will run this code line by line
- The first line has details we'll cover in later lectures
  - int is the return type
    - int is short for integer
  - main is the name of the function
  - (void) means the function takes no input

```c
// A demo program
// Welcome to COMP1511 :)

#include <stdio.h>

int main(void) {
    printf("Welcome to COMP1511\n");
    return 0;
}

```

Breaking It Down

- Between the { and } are a set of program instructions
- printf() makes text appear on the screen
  - This is a function that comes from stdio
- return is a C keyword
  - Delivers the output of the function



```c
// A demo program
// Welcome to COMP1511 :)

#include <stdio.h>

int main(void) {
    printf("Welcome to COMP1511\n");
    return 0;
}

```

Editing and Compiling

- In the Linux terminal we will open the file to edit
  - `code hey.c`
- Once we're happy with what we've written, we can compile it
  - `dcc hey.c -o hey`
    - The `-o` tells the compiler to make a program called hey that we can then run
- The `./` command lets us run the program in our current directory
  - `./hey`

And We’re Off!

- Try it yourself
- Try it using VLab on your own computer
- Try setting up a program environment on your own computer

Questions?

- Check the course forum
- Email us
  - cs1511@unsw.edu.au
- We are here to help!

TOMORROW IS ONLINE ONLY PLEASE DO NOT FORGET THAT

Questions?

See You Tomorrow!

Questions?