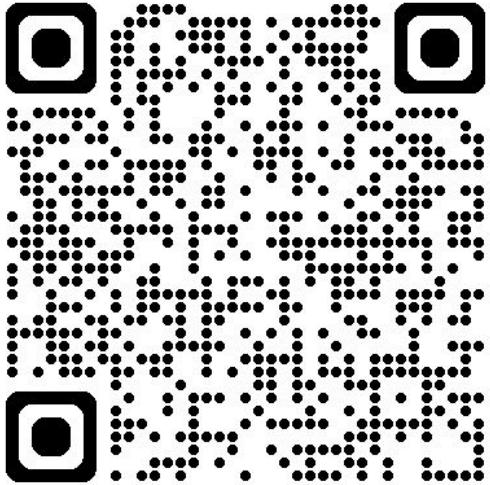
COMP1511/1911 Programming Fundamentals

Week 10 Lecture 1

Revision

Link to Week 10 Live Lecture Code

https://cgi.cse.unsw.edu.au/~cs1511/25T3/code/week_10/



My Experience Surveys

http://myexperience.unsw.edu.au/

Revision Sessions

- Last set of revision sessions on in week 11
- Look out for announcement and sign ups on the forum soon

Week 10 Practice Exams

- Held in Labs for 2 hours
- This is how you get lab marks for week 10
  - Marks are based on attempting it.
- If you are in an online tut-lab
  - access code “COMP1511”
  - sign up for an in-person lab for week 10 https://buytickets.at/comp1511unsw/1857368
- Can be done online, but don’t miss another chance get used to the exam environment.
- Exam environment walkthrough https://youtu.be/O05-z35PJXQ

Week 10 Practice Exams

During the practice exams:

- Any autotests or submissions will not be stored on student history
- Any code written in the exam environment will not be accessible to you once you log out of the exam environment.
- Practice exam questions will also be available at home via the week 10 lab on the course website

The Exam: Time and Date and Location

- Date: Wednesday 3rd December
- 3 hours + 10 minutes reading time
- There will be 2 sessions of the exam.
  - **Wed 03/12/2025 Morning** (09:45-13:00)
  - **Wed 03/12/2025 Afternoon** (12:50 corralling 13:25 - 16:40)
- Register your preference for your exam time here:
  - https://cgi.cse.unsw.edu.au/~exam/25T3/seating/register.cgi
- The exact time and location of your exam will be available mid Week 11 - you will receive another email with instructions once it is available.

Last Week

- Email Management System Linked List Application
- Exam Details

Today's Lecture

- Revision:
  - Linked List Hurdles
  - Array Hurdles
- Special Guest Tom Kunc from Optiver
  - past CSE student/COMP1511 tutor and admin
  - Array problem solving coding example
    - Real world (Optiver) application

Sample Hurdle Exam Questions

Linked List Hurdle Example (●○○)

`linked_list_largest.c`

Perform some computation or comparisons on a linked list

Given a linked list, return the largest value in that list.
If the list is empty, return -1

```
int largest(struct node *head);
```

Linked List Hurdle Example (●○○)

linked_list_difference.c

Perform some computation or comparisons on 2 linked lists

Given 2 linked lists return the difference in the number of nodes in each list.

```
int difference(struct node *head1,
               struct node *head2);
```

Linked List Hurdle Example (●○○)

`linked_list_equal.c`

Given two linked lists, return the number of values in the first linked list that are equal to the corresponding values in the second linked list.

```
int equal(struct node *head1,
          struct node *head2);
```

Array Hurdle Example (●○○)

`array_count_multiples.c`

Given an array:
Count the number of multiples of 3 in the array.

```
int array_count_multiples (int size,
                           int array[]);
```

Array Hurdle Example (●○○)

array_step_count.c

An array of structs of type:

```
struct steps {
    int number;
    char direction;
};
```

return the total of the number of steps taken in a given direction

```
int step_count(char dir, int size,
               struct steps array[]);
```

Array Hurdle Example (●○○)

array_struct_capitals.c

An array of structs of type:

```
struct initials {
    char first_initial;
    char last_initial;
};
```

return the count of all structs that don't have capital letters for both of their initials.

```
int capitals(int size, struct initials array[]);
```

Linked List Hurdle Example (●●◌)

`linked_list_duplicate.c`

Duplicate every node in the list by inserting the same node after the original node.

```
struct node *duplicate(struct node *head);
```

Linked List Hurdle Example (●●◌)

`linked_list_delete_duplicate.c`

Delete the first instance of a duplicate in the given list e.g.

1->3->5->3->1->7->X would give
1->3->5->3->7->X

```
struct node *delete_duplicate(struct node *head);
```

Array Hurdle Example (●●◌)

array_multiplied.c

Write a C program that reads integers into an array from terminal until a number is entered which when multiplied by another number previously entered results in 56. E.g.

```
./array_multiplied
2
3
28
28 * 2 = 56
```

Array Hurdle Example (●●◌)

array2d_min_row_sum.c

Write a C program that finds the sum of the minimum numbers in each row in a 2D array.

```
int min_row_sum(int size,
                int array[NUM_ROW][NUM_COL]);
```

Guest Speaker: Tom Kunc

- Then
  - **CSE student** from 2018-2022
  - tutor for COMP1511, COMP1521, and COMP6991;
  - course admin for COMP1511;
  - helped write "the Rust course" (COMP6991).
- Now
  - works at **Optiver** as a team lead of the Research Platform group
  - still teaches for UNSW in the Rust course;
  - is learning to scuba dive.

Guest Speaker From Optiver Tom Kunc

- Talking about two topics:
  - COMP1511/1911 skills used in industry
  - Array revision problem
    - extra problem solving practice for the exam
    - exploring a typical problem Optiver has had to solve.

The Final Lecture on Tuesday

- The final Kahoot
- Continue with revision
- Q&A with some course tutors
  - Exam tips
  - What to enrol in next term
  - Getting internships and jobs
  - Being a tutor

Feedback Please!

Your feedback is valuable!

If you have any feedback from today's lecture, please follow the link below or use the QR Code.

Please remember to keep your feedback constructive, so I can action it and improve your learning experience.

https://forms.office.com/r/Zp4DFZW35r

Reach Out

Content Related Questions:

Forum

Admin related Questions email:

cs1511@unsw.edu.au