

#### **COMP1511 PROGRAMMING FUNDAMENTALS**

## Lecture 18

**Revision:** Pointers and Linked Lists





COMP1511 Programming Fundamentals



 Talked about the exam Revised Strings/Char

### YESTERDAY...



COMP1511 Programming Fundamentals

- Pointers
- Linked Lists



# WHERE IS THE CODE?

LIVE LECTURE CODE CAN BE FOUND HERE:

https://cgi.cse.unsw.edu.au/~cs1511/21T3/live/Week10/

### HOUSEKEEPING

#### **MY EXPERIENCE SURVEYS HTTP://MYEXPERIENCE.UNSW.EDU.AU/**



Tell us about your experience and shape the future of education at UNSW.

**Click the link in Moodle** 

Please be mindful of the UNSW Student Code of Conduct as you provide feedback. At UNSW we aim to provide a respectful community and ask you to be careful to avoid any language that is sexist, racist or likely to be hurtful. You should feel confident that you can provide both positive and negative feedback but please be considerate in how you communicate.







### REVISION **POINTERS**

- Pointers are another variable type in C
- Pointers store the memory address of another variable & – gives the address of

  - \* dereferences a pointer, so provides the value of stored at the address the pointer is at



# **REVISION POINTERS**

}

```
int main(void) {
    int a = 5;
    int b = 10;
    int *ptr1;
    int *ptr2;
    ptr1 = &a;
    ptr2 = &b;
    *ptr1 = 10;
    ptr1 = ptr2;
    *ptr1 = 20;
    printf("a = %d\nb = %d\n", a, b);
    return 0;
```

DXAAA DXAAA OX BBB OX CCC PTV 1

• Let's see an example:

#### pointer.c



## REVISION

### **POINTERS**

• Let's see an example:

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

```
int sumPlusTwo(int *number1, int *number2){
    return *number1 + *number2 + 2;
void numPlusOne(int *number1){
    *number1 = *number1 + 1;
void swap (int *number1, int *number2){
    int temp = *number1;
    *number1 = *number2;
    *number2 = temp;
int main(void){
    int num1 = 13;
    int num2 = 1;
    swap(&num1, &num2);
    numPlusOne(&num1);
    numPlusOne(&num1);
    return 0;
```

printf("%d + %d + 2 = %d\n", num1, num2, sumPlusTwo(&num1, &num2));

printf("%d + %d + 2 = %d\n", num1, num2, sumPlusTwo(&num1, &num2));

#### pointer2.c



Did you enjoy your first taste of programming?





#### 5**)** amming?



### REVISION **LINKED LISTS**

Some special boundary conditions that you need to consider when you manipulate lists:

- Empty list
- List with 1 element
- list
- etc)

Something happening at the beginning of the

 Something happening at the end of the list • Something will not occur, the item is not in the list (inserting after a number that doesn't exist

## REVISION LINKED LISTS

linked\_list.c linked\_list.h main.c **Problem 1:** Find all the elements in the linked list that are divisible by a specified number and output all of these numbers

Now delete all the nodes that are divisible by the specified number

**Problem 2:** Find the range (the difference between the biggest term and the smallest term) of a linked list

**Problem 3:** Insert the a number into the linked list in the middle of the linked list. Assume that there is always going to be an even number of numbers in the list before insertion

### **THANK YOU**



Thank you all so much for tuning in, for learning, for engaging, and I hope that you had an enjoyable intro to programming. Don't forget that Rome wasn't built in a day, and becoming a better programmer entails lots of practice!

I really appreciate the engagement that you have shown throughout the lectures, and I wish you all well in the final exam.

Have a wonderful break over the summer holidays, I hope you get to see your friends and family and enjoy some down time.

Good Luck for your future courses, and I may see some of you again :)

### WHAT DID WE LEARN TODAY?



pointer.c pointer2.c

#### LINKED LISTS

linked\_list.c main.c linked\_list.h

### ANY QUESTIONS? DON'T FORGET YOU CAN ALWAYS EMAIL US ON CS1511@CSE.UNSW.EDU.AU FOR ANY ADMIN QUESTIONS

PLEASE ASK IN THE FORUM FOR CONTENT RELATED QUESTIONS

