COMP1511 PROGRAMMING FUNDAMENTALS

LECTURE 15

Exam Information

Multi-file projects

LAST TIME...

- What is a Linked List?
- Linked List traversing
- Linked Lists inserting
- Linked Lists deleting

THIS SHORT WEEK...

- Exam info!
  - Format
  - Preparation
  - Hints and tip
- Multi-file projects

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

Live lecture code can be found here:

HTTPS://CGI.CSE.UNSW.EDU.AU/~CS1511/23T1/LIVE/WEEK09/

COURSE FEEDBACK

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

my Experience

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.

UNSW SYDNEY

my Experience surveys
http://myexperience.unsw.edu.au/

THE EXAM

WHAT IS
IN IT?

- Everything that we have learnt so far
  - Lots of focus on:
    - Simple IF statements and WHILE loops
    - Variables: int, double, char, structs
    - Strings
    - Arrays
    - Pointers
    - Linked Lists

THE EXAM

TIME/DATE

- Date: 2nd May and 3rd May (choice of four sessions with two sessions running each day)

- There will be FOUR sessions of the exam. The students sitting the exam in the afternoon will be corralled for a period of twenty-thirty minutes as we conduct the changeover. The exam will be in different CSE lab locations, and you will be given a seating arrangement closer to the date of the exam, so you know where you need to be and at what time. The sessions are:

- Morning: 10:15-13:30

- Corralling: 13:20-13:55 (although arrive at 13:20, corralling doors can close at 13:30)

- Afternoon: 13:55-17:10

THE EXAM

- The lab NEXT week will provide you with a test environment that will be similar to your exam - this will allow you to familiarise yourself with the setup
  - Give is the same as in your labs/weekly tests
  - Autotests are run the same way etc
  - Submit as many times as you want - only last submission will be marked

THE EXAM

EXAM HURDLES

- There's an array hurdle, question 5 or 7
  - You must earn a mark of 50% or more in at least one array hurdle question
- There's a linked list hurdle, question 6 or 8
  - You must also earn a mark of 50% or more in at least one linked list hurdle question

THESE QUESTIONS WILL BE CLEARLY MARKED ON THE EXAM AS HURDLES

THE EXAM

EXAM CONDITIONS

- Close book, however:
  - Our course website will be open, which means you will have access and are allowed to refer to:
    - Any lecture material or code
    - Any tutorial material or code
    - Any laboratory material or code
  - Exam conditions still apply!!!
    - Do not communicate with anyone about the exam within 24 hours of the exam start time - this is considered plagiarism
  - No discussion of the exam or sharing your code with anyone except for COMP1511 staff

THE EXAM

- If you experience any issues during the exam, please raise your hand and wait for an invigilator.

THE EXAM

- When you come into the room and seat yourself, there will be instructions provided to you on how to start the exam
  - We personalise the papers, so your paper may be different from that of someone else.
  - The different sessions also have different exam papers
  - The command needed to fetch your exam. You will use a similar command in your practice exam to get you used to the process:

```
File  Edit  View  Terminal  Tabs  Help
avas605@vx6:~$ 1511 fetch-exam
```

THE EXAM

FIT TO SIT

Fit to Sit Policy:

https://www.student.unsw.edu.au/exam-rules

By sitting the exam on the scheduled assessment date, you are declaring that you are fit to do so and cannot later apply for Special Consideration.

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

THE EXAM

FIT TO SIT

If, during an online exam you feel unwell to the point that you cannot continue with the exam, you should take the following steps:

- Stop working on the exam and take note of the time
- Please raise your hand and let the invigilator know you are unwell.
- Immediately submit a Special Consideration application saying that you felt ill during the exam and were unable to continue
- You must provide a medical certificate dated within 24 hours of the exam, along with any record of the conversation you have had with us (basically that you have let us know you feel unwell before leaving)

THE EXAM

SUPP EXAM

- If you are granted special consideration based on the Fit to Sit university policy, a supplementary exam will be held in May of 2023. If you think you will need to sit this exam, make sure you are available.
- Fit to Sit Policy: https://www.student.unsw.edu.au/exam-rules

THE EXAM

WEEK 11 REVISION CLASSES

In Week 11, we will be running some revision classes.

Registration and details will be announced soon...

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

THE EXAM

WEEK 10 - FLIPPING THE CLASSROOM

In Week 10 Lecture slots, we will be flipping the classroom! This means, we will have a series of problems that will be available for you to work through - and you will vote on problems that we will pseudocode and solve together and then also have time to work on problems in small groups. We will also run some fun Kahoots during the time (alright fine, fake Kahoots)

When: Monday 11-1pm lecture

When: Thursday 12-2pm lecture

THE EXAM

FORMAT

The format of the exam will be:

- 11 Practical Answer Questions with the following distribution of marks:
  - Q1 - Q4 (5 marks each) They require you to understand what pieces of code are doing, interpreting code, interpreting diagrams, etc.
  - Q5 - Q8 (12 marks each) Programming questions
  - Q9 - Q10 (11 marks each) Programming questions
  - Q11 (10 marks each) Programming questions
  - Similar in style to the questions you did in your problem sets, revision exercises
  - These are also rated to give you an idea of how difficult each question is.
- We hope that everyone can attempt and complete the first eight questions

THE EXAM

Q1-Q4

Practical Application Answer Questions

Number of questions: 4

Marks: 5 mark each (total of 20 marks)

- These questions will be about whether you understand
  a coding concepts and the C programming language
- Some are: "What will this code do and what values
  does it take?"
- Some are: "How does this concept work?"
- Some are: "Debug the following code to make sure it
  works to produce the desired output."
- Some examples are in the Week 10 Prac Exam

THE EXAM

PRACTICAL QUESTIONS

Practical Questions

Number of questions: 7

Marks: varied

Q5 - Q8 (12 marks each)

Q9 - Q10 (11 marks each)

Q11 (10 marks) (total of 70 marks)

- Questions are similar to the Revision Exercises and Problem Sets

  - Stages of difficulty from basic to very hard (will be marked in the same rating system as your problem sets)

  - Some will have provided code as frameworks

  - Each question will need to be written, compiled and tested

  - You will have access to autotests (but they're just tests!)

THE EXAM

PRACTICAL QUESTIONS

Practical Questions

Number of questions: 7

Marks: varied

Q5 - Q8 (12 marks each)

Q9 - Q10 (11 marks each)

Q11 (10 marks) (total of 70 marks)

- Harder questions will have less autotests

- There will be no specific style marking, so you don't need to explain your code in comments but it needs to be readable so I can mark it!

THE EXAM

PRACTICAL QUESTIONS

- When you fetch your exam in the beginning, it will also copy over any starter files you may need for the practical questions.

- Read all the questions before starting

- Start with the easier questions

- Prepare! A couple of minutes thinking and drawing a diagram will clarify how you're going to approach a question

- Use your problem set/revision practice! Debugging and testing will be important here

- Less questions answered completely is better than more questions partially answered

THE EXAM

PRACTICAL QUESTIONS

QUESTION 5 AND QUESTION 6

These are: ●○○

- Similar to Practical test question 5 or 6
- Question 5 is an array hurdle
- Question 6 is a linked list hurdle

- Tests your ability to:
  - Create simple C programs
  - Use variables (int, double, char)
  - Use scanf and printf
  - Use IF statements and WHILE loops
  - Use of simple structs
  - Use of arrays of int/double/struct in Q1
  - Use of linked list of ints/doubles (no insertion or removal of nodes) in Q6

THE EXAM

EXAMPLE QUESTION 5

● ○ ○

- Loop through an array of structs and gather some kind of information

Given an array of structs, where each struct is:

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

Print out the total of the number of steps taken in a specific direction. So for example, if direction is 'l', find all the structs with direction as 'l' and add the numbers in those structs up. Edit the function

```
int total (int size, struct direction array[MAX])
```

THE EXAM

EXAMPLE QUESTION 6

● ○ ○

Perform some computation on a linked list

Given a linked list, print the largest value in that list

Edit the function

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

THE EXAM

PRACTICAL QUESTION 7 AND QUESTION 8

●●○

These are: ●●○

- Similar to Practical test question 7 and 8
- Question 7 is an array hurdle
- Question 8 is a linked list hurdle

If you have answered Q5 and Q6, this means that you have already passed the hurdles of the exam

- These are harder applications of the hurdles
- You will need to know everything from Q5 and Q6, in addition to:
  - Looping through more than once (maybe)
  - Some insertion/removal of nodes in Q8
  - Testing more difficult conditions and keeping track of more than one thing.

THE EXAM

PRACTICAL QUESTION 9 AND QUESTION 10

●●●

These are: ●●●

- Harder manipulation of arrays (Q9)
  - Possibly fgets or string manipulation
- Manipulate linked lists (adding and removing items etc) (q10)
  - Potentially use malloc() and free() with structs and pointers
- Again, more complex combinations, and some questions requiring interesting problem solving

THE EXAM

PRACTICAL QUESTION 11

COMBINATION OF:

●●● 💀

This one: ●●● and 💀

- For those aiming for a HD mark
- Everything taught in the course might be in these questions
- Think "Exercises", even some of the hard ones!
- Will also test your ability to break a problem down into its parts
- The Prac Exam has an example of past Question 11 so you can see the difficulty level
- Partial completion of this question will award some marks

THE EXAM

WHAT SHOULD I STUDY?

The basics are important!

- Know how to use both arrays and linked lists
- Go back and do the problem sets if you haven't already
- The revision exercises on the course webpage are also very useful

- Variables, Structs, enums, IF, Looping, Functions, Arrays, Linked Lists are very important to understand!
- You will need to have some understanding of Strings, Pointers, and Memory Allocation to be able to work successfuly with char arrays, and linked lists

THE EXAM

MARKING

- Most of the marking will be automated
- Make sure your input/output format matches the specification
- Answers for hurdles will also be checked by hand
- Marks will be earnt for correct code, not for passing autotests
- Minor errors, like a typo in an otherwise correct solution, will only result in a small loss of marks

THE EXAM

YOU'VE GOT THIS

- Whilst some parts of the exam (the later questions) have been designed to be very challenging, you do not need to complete them to be successful in getting a great mark in this subject
- Make sure you breathe!

THE EXAM

YOU'VE GOT THIS

- When you are struggling to understand a question (particularly linked lists) = DRAW DIAGRAMS!

- Go over your problem sets and revision questions for extra practice.

- Revision classes will be run in Week 11 - details coming

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

BREAK TIME...

How are you all doing?

Are you doing anything fun once the first term is over?

Are you excited to be in computing and to continue to build your knowledge further?

MULTI FILE PROJECT

WHAT ARE THEY?

- Big programs are often spread out over multiple files. There are a number of benefits to this:
  - Improves readability (reduces length of program)
  - You can separate code by subject (modularity)
  - Modules can be written and tested separately
- So far, we have already been using the multi-file capability. Every time we #include, we are actually borrowing code from other files
- We have been only including C standard libraries

MULTI FILE PROJECT

WHAT ARE THEY?

- You can also #include your own! (FUN!)
- This allows us to join projects together
- It also allows multiple people to work together on projects out in the real world
- We will also often produce code that we can then use again in other projects (that is all that the C standard libraries are - functions that are useful in multiple instances)

MULTI FILE PROJECT INCLUDES

.H FILE
.C FILE (MAYBE MULTIPLES)

- In a multi file project we might have:
- (multiple) header file - this is the .h file that you have been using from standard libraries already
- (multiple) implementation file - this is a .c file, it implements what is in the header file.
- Each object or file that you write, will have its own implementation file
- a main.c file - this is the entry to our program, we try and have as little code here as possible

header file

#include "   .h"

impleme ntation file

.c

HEADER FILE

#INCLUDE "SOMETHING.H"

Typically contains:

- function prototypes for the functions that will be implemented in the implementation file
- comments that describe how the functions will be used
- #defines

- the file basically SHOWS the programmer all they need to know to use the code
- NO RUNNING CODE
- This is like a definition file

IMPLEMENTATION FILE

SOMETHING.C

This is where you implement the functions that you have defined in your header file

IMPLEMENTATION FILE

MAIN.C

This is where you call functions from that may exist in other modules.

AN EXAMPLE

A MATHS

- We will have three files:
  - header file - maths.h
  - implementation file - maths.c
    - #include "maths.h"
  - main file - main.c
    - #include "maths.h"

```
*maths.h
1 // This is the header file for the maths module example
2 // The header file will contain:
3 //     - any #define
4 //     - function prototypes and any comments
5
6 #define PI 3.14
7
8 //Function prototype for a function that calculates
9 //square of a number
10 int square(int number);
11
12 //Function prototype for a function that calculates
13 //sum of two numbers
14 int sum(int number1, int number2);
15
```

```
main.c
1 //This is the main file in our program
2 //This is where we drive the program from and where we
3 //make calls to our modules. We need to include the
4 //header file for each module that we want to use functions
5 //from
6
7 #include <stdio.h>
8 //Include the header file:
9 #include "maths.h"
10
11 int main (void) {
12     int number = 13;
13     int number2 = 10;
14
15     printf("The square of the number %d is %d\n", number, square
(number));
16     printf("The sum of %d and %d is %d\n", number, number2, sum
(number, number2));
17     return 0;
18 }
```
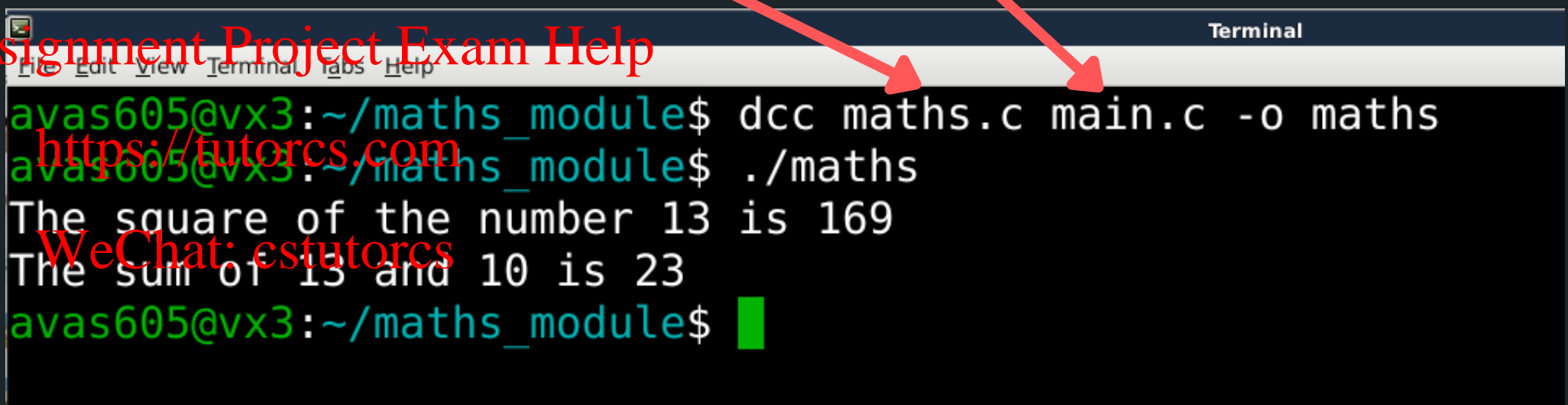
```
maths.c
1 //This is the implementation file of maths.h
2 //We defined two functions in the header file,
3 //and this is where we will implement these two
4 //functions
5
6 //Include your header file in the implementation file
7 //by using the below syntax
8
9 #include "maths.h"
10
11 int square(int number) {
12     return number * number;
13 }
14
15 int sum(int number1, int number2) {
16     return number1 + number2;
17 }
```

COMPILING A MULTI FILE

COMPILE ALL C FILES IN THE PROJECT

- To compile a multi file, you basically list any .c files you have in your project
  - In the case of our example, we have a maths.c and a main.c file):

```
Terminal
File Edit View Terminal Tabs Help
avas605@vx3:~/maths_module$ dcc maths.c main.c -o maths
avas605@vx3:~/maths_module$ ./maths
The square of the number 13 is 169
The sum of 13 and 10 is 23
avas605@vx3:~/maths_module$
```

- The program will always enter in main.c, so there should only be one main.c when compiling

WHAT DID WE LEARN TODAY?

EXAM

Details, details, details!

MULTI-FILE PROJECTS

maths.c

main.c

maths.h

linked_list.c

linked_list.h

main.c

REACH OUT

CONTENT RELATED QUESTIONS

Check out the forum

ADMIN QUESTIONS

cs1511@unsw.edu.au