• Automarking (80%) available
• andrewt browsed submissions from best point scores down
• picked point score for HD, point score DN, .... based on code around that score
• should be inline with assignment spec
• tutor can adjust if obviously unfair
• very low score/doesn’t compile - tutor manually assesses
• tutors will start style marking soon, but they have exams
Assignment 3

- will be run in various single-bot worlds
- will be run with all other student’s bots in huge multi-bot worlds
- similar assessment to assignment 2
- tutor can adjust if obviously unfair
- very low score/doesn’t compile - tutor manually assesses
- performance mark (80%) by end of stuvac
- style marking soon, will be delayed by tutors exams
Lab Week 13

- Complete myExperience
- Practice exam - very important
- All versions of practice exam questions and solutions released next week
- Get help with assignment 3
Assessment

- 12% Labs
- 8% Weekly tests
- 6% Assignment 1
- 12% Assignment 2
- 12% Assignment 3
- 50% Final Exam (15/06)

Marks might be scaled to ensure an appropriate distribution, Sometimes overall mark scaled up a little. Occasionally component marks scaled up. Scaling down almost never done
To pass the course, you must do all of these:

- score 50/100 overall
- solve problem using arrays in the final exam
- solve problem using linked-lists in the final exam
- questions for each hurdle clearly labelled in final exam
Labs (12% of final mark)

- Each lab mark worth 1.3.
- Best 10 lab marks summed.
- Multiplied by 13.2/13
- Capped at 12 marks.
- Current total (without week 13) visible in class database

Original assessment was best 11 of 12 labs each worth 1.2
But changed because we only had 11 labs!
Hence multiplication by 13.2/13
Weekly Tests (8% of final mark)

- Each weekly test worth 1 mark
- Best 8 test marks summed.
- Current total (without week 12 test) visible in class database
Final Exam (50% of final mark)

- Final exam held in 2 sessions on Friday 15/06 in CSE labs
- If you have another exam on 15/06 you will be automatically allocated to a non-clashing session.
- Email will be sent Wednesday telling you how to indicate a preference for exam session
- As many students as possible allocated their preferred session.
- If you have religious or other important reason please first indicate preference, email cs1511 if preference not met.
- Your exam time and location posted to class webpage
- 3 hours closed book exam - no materials allowed.
- You will be able to use an attendance sheet for rough work
- Exam has 2 parts - do both of them
- Exact format (skeleton exam) released by 13/06
Exam Part 1

Must be completed during 1st 30 minutes of 3 hour exam. No use of computer allowed during this part except to enter answers into application and view online documentation, You can not run terminal or dcc or gcc or clang or ....

- Probably about 15 questions
- Some questions will ask you to read code and indicate what it does.
- Questions will be short answer or multiple choice
- Practice exam questions - good guide to what to expect (but harder)
Exam - Part 2

- 7-8 questions
- Most questions will describe a task and ask you to write a program or function
- Questions will usually include examples.
- Question may give you some starting code
- You may or may not be given test data or other files
- Most or all will have autotests - passing autotest does not guarantee marks. Do your own testing.
- It is not sufficient to match any supplied examples.
- You must use C to answer the question.
- Can read questions in first 35 minutes.
- Can not run editor/dcc in first 35 minutes.
Question 1-2 will be easier questions. Similar difficulty to first question on weekly tests

- create a simple C program
- declare and use int & double variables
- use scanf to input ints or double
- use print to output ints or double
- write if statements
- write loops, including nested loops
- access command line arguments and convert to int or double
- use arrays to store ints/doubles
Read 1 or more values and then do some computation, e.g:
Your program should read two ints. It should then print a line for all the even numbers that lie between these 2 values. The line should be the even number and its square.

% a.out
Enter lower: 12
Enter upper: 17
14 196
16 256
Perform some computation from command line arguments, e.g.:
Your program will be given 1 or more command line arguments which you can assume are all integers, calculate the sum of their squares and print this.

% a.out 5 4 3
50
Question 3-4

Similar difficulty to q2 on weekly tests
You need to be able to

• use fgets to read lines & fgetc to read chars
• read until end-of-input using scanf, fgets, fgetc
• use arrays to store strings
• manipulate strings
• do computations on linked lists

Your revision should include weekly tests, all tutorial questions and all standard lab exercises
Question 5-6

Similar difficulty to q3 on weekly tests, harder standard lab exercises or easier challenge exercises.
You need to be able to

- read & write files
- malloc
- change strings
- change linked lists

Your revision should include easier challenge exercise.
Difficult questions for HD students. Complex programming using any of the features covered in course. Your revision should include all challenge lab exercises (except silly puzzles in early weeks)
Hurdle Requirements

To pass the course

- solve problem using arrays in the final exam
- solve problem using linked-lists in the final exam
- There will be at least two questions for each hurdle requirement.
- At least one of the question for each hurdle will be earlier in the exam among easier questions.
- Good strategy - to do get hurdles out of the way.
Your answers will be run through automatic marking software.

Please follow the input/output format shown exactly.

Please make your program behave exactly as specified.

All answers are also hand marked. The automatic marking is to assist these markers.

No marks awarded for style or comments.

Use decent formatting so the marker (and you) can read the program.

Comments only necessary if you want to tell the marker something.

Minor errors will result in only a small penalty.

E.g. an answer correct except for a missing semi-colon would receive almost full marks.

No marks will given unless an answer contains a substantial part of a solution (30+%).

No marks just for starting a question and writing some C
No past papers are available.
No past exam suitable guide.
All 30 weekly test questions chosen to be like exam questions
By attending the exam, you are saying ”I am well enough to sit it”. If you really are sick, stay home and apply for Special Consideration. Applications for Special Consideration from people who sat the exam will be ignored. If you become ill during the exam, ask the supervisor to contact lecturers and then talk to lecturers.
Provisional results will be made available via the class database when marking is complete. 1000 exams take a long time to mark - so probably July 02. We’ll send email announcing this. You will be emailed time(s) which you can view your exam and check marking. Final results will appear on myUNSW.
Students will be offered a supplementary exam if they miss the original exam due to (documented) illness or misadventure. Also automatic supp if your mark is 40-49 and have attended 9+ labs, 8+ tests and reasonable attempts on assignments. Also automatic supp if your mark is 50+ but you fail a hurdle. The supp tentatively scheduled for Friday 20 July. Your responsibility to be available - no alternative!
Supplementary Exam

Similar format to final exam (no skeleton released).
Supplementary exam tentatively scheduled for Friday 20 July.
There is no alternative to the supplementary exam - if you miss it your grade will be FL.
Don’t email me asking to have the supplementary at another time.
If you think you might be offered supplementary assessment, make sure you are available that week.
Supplementary assessment offers will be sent by email.
Need to get students with programming experience writing programs in week 0 (or before)
Inefficient use of lecture time (in stream A at least).
More integration of tools (like git) needed.
Sudden jump in lab-difficulty before HECS census date.
Tutors were great.
Students were great.
Most students seem to have learned a lot from labs, tests and assignments.
Please give feedback on all your courses via myExperience. We’ll circulate a COMP1511 specific survey later asking questions like:

- should we keep pair programming?
- should we have mid-session exams?

Please complete it
Good Luck in the exam.
I hope you get the COMP1511 mark you deserve.
I know many of you have worked very hard.
I hope you have been rewarded with an understanding of computers & programming that will help you do interesting and important things in future.