Week 13: Revision

COMP2129

Couse review, Exam, future

- Postconditions of the course
- Preparing for the Exam
- Soothing examples
- Stats & stuff

Key postconditions

- Ability to read and write correct, clean code in C that allocates, deallocates and manages memory
- Understanding of common memory-related errors (such as memory leaks, dangling pointers) and how to avoid these.
- Ability to correctly implement standard linked list data structures

Key postconditions

- Ability to read and write code that correctly uses the main standard library functions, especially for I/O, file handling, and string handling.

- Ability to use code quality strategies appropriate for C, including preprocessor techniques, and use of common idioms

- Experience in using debugging tools.

Key postconditions

- Experience of following a thorough automated testing regime using tools such as make, diff, scripts to present the outcomes, and a tool to manage regression testing.
- Understanding of the approach and concepts of Unix, including its tools philosophy, processes (including pipes and redirection), the file system, and the shell.
- Ability to learn to use Unix commands and system calls (including usage of flags etc) from online manual system.

Key postconditions

- Understand the limitations of sequential computing performance and concepts in parallel programming
  - task-parallelism, data-parallelism
  - Synchronisation
  - Deadlock, livelock, starvation
  - Locking hierarchy
  - Scheduling
  - …
- Ability to design a parallel solution to a given problem. Use a parallel thread library such as Pthreads to derive the code.
- Ability to measure performance through profiling and identify load balancing issues or bottlenecks in either software and hardware.

Essentials for the pass

- C aspects that are conceptually like Java
- C pointers basics
- C pointers with arrays and strings
- C pointers with lists and other data structures
- C memory model
- Unix basics

Preparing for core

- Review the lab material, lectures
- Especially review C aspects
- Practice all the UNIX from labs and lectures
- Write small code examples
  - Get them to work
  - Test them (write small test scripts)
  - So you know they are right
  - Draw pictures of memory

Preparing for core

- Aim to write clear code
- Write comments to make your thinking clear
    - Eg. Initialise Arr to values 1 .. N.
- Draw pictures
    - They really do help
    - Good programmers draw them – So why not you?

- Map out solution at high level
    - Write pseudo-code if it is easier first
    - Write comments for each main step
    - Write actual code or function call
    - Later, write within each of the function

Example

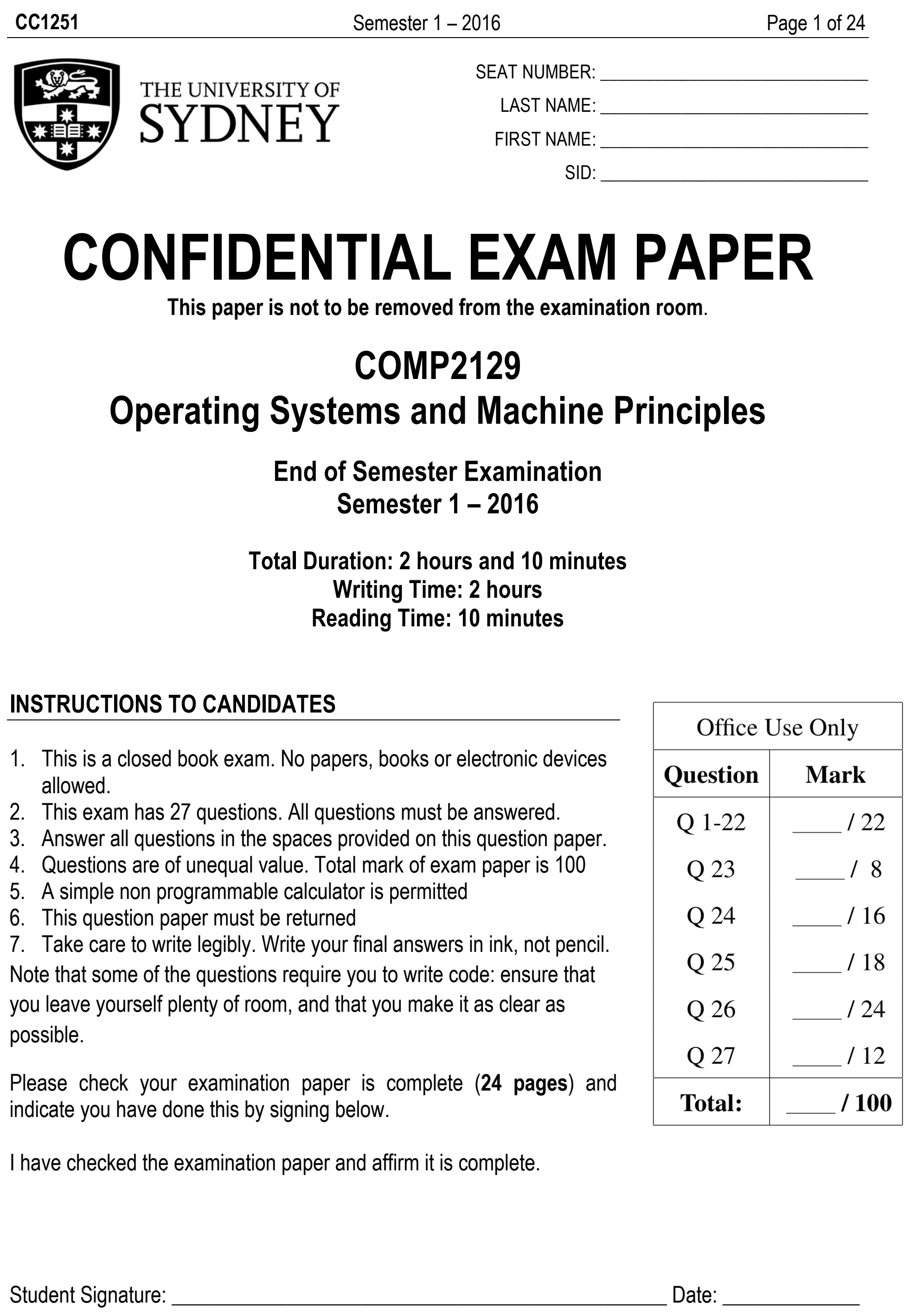
- Context: there is a linked list of ints:
  - Task: add 1 to the value of each element
- Steps:
  - Needs to traverse a list
  - Needs to have a list that has been built
- Examples you know you need to test
  - Empty list
  - Single element list
  - Problems at head
  - Problems at end
  - “Normal “case

Example

- Start with diagram/pictures, whatever helps you
- Write an outline for special cases
- Write code for special cases

- Think about top level steps
- Write a main() to reflect this
  - Comments
  - Function calls
  - Flesh out details

- Aim for clarity, simplicity, correctness
- Once high level is done, write each function



CC1251 Semester 1 – 2016 Page 1 of 24

THE UNIVERSITY OF SYDNEY

SEAT NUMBER: ______________________

LAST NAME: ______________________

FIRST NAME: ______________________

SID: ______________________

CONFIDENTIAL EXAM PAPER

This paper is not to be removed from the examination room.

COMP2129
Operating Systems and Machine Principles

End of Semester Examination
Semester 1 – 2016

Total Duration: 2 hours and 10 minutes
Writing Time: 2 hours
Reading Time: 10 minutes

INSTRUCTIONS TO CANDIDATES

1. This is a closed book exam. No papers, books or electronic devices allowed.
2. This exam has 27 questions. All questions must be answered.
3. Answer all questions in the spaces provided on this question paper.
4. Questions are of unequal value. Total mark of exam paper is 100
5. A simple non programmable calculator is permitted
6. This question paper must be returned
7. Take care to write legibly. Write your final answers in ink, not pencil.

Note that some of the questions require you to write code: ensure that you leave yourself plenty of room, and that you make it as clear as possible.

Please check your examination paper is complete (**24 pages**) and indicate you have done this by signing below.

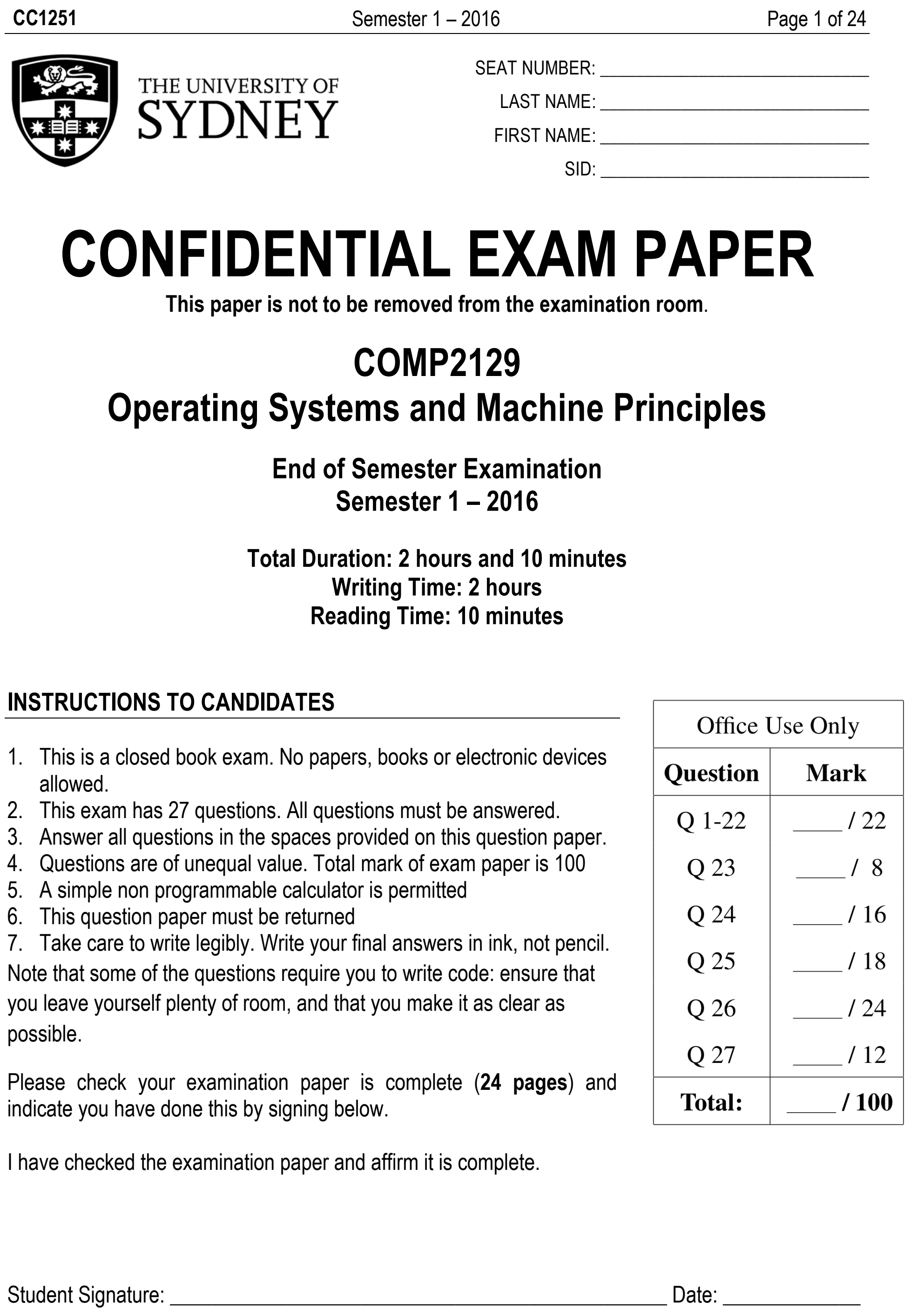
I have checked the examination paper and affirm it is complete.

Office Use Only	
Question	**Mark**
Q 1-22	____ / 22
Q 23	____ / 8
Q 24	____ / 16
Q 25	____ / 18
Q 26	____ / 24
Q 27	____ / 12
Total:	____ **/ 100**

Student Signature: ______________________ Date: ____________

Front page

- Name
- Write your SID
  - Don't forget!
- Multiple choice multiple answer
- CLOSED BOOK
  - NO A4 sheet
- Non programmable calculators
- 24 pages. Time planning essential



CC1251 Semester 1 – 2016 Page 1 of 24

THE UNIVERSITY OF SYDNEY

SEAT NUMBER: ______

LAST NAME: ______

FIRST NAME: ______

SID: ______

CONFIDENTIAL EXAM PAPER

This paper is not to be removed from the examination room.

COMP2129
Operating Systems and Machine Principles

End of Semester Examination
Semester 1 – 2016

Total Duration: 2 hours and 10 minutes
Writing Time: 2 hours
Reading Time: 10 minutes

INSTRUCTIONS TO CANDIDATES

1. This is a closed book exam. No papers, books or electronic devices allowed.
2. This exam has 27 questions. All questions must be answered.
3. Answer all questions in the spaces provided on this question paper.
4. Questions are of unequal value. Total mark of exam paper is 100
5. A simple non programmable calculator is permitted
6. This question paper must be returned
7. Take care to write legibly. Write your final answers in ink, not pencil.

Note that some of the questions require you to write code: ensure that you leave yourself plenty of room, and that you make it as clear as possible.

Please check your examination paper is complete (**24 pages**) and indicate you have done this by signing below.

I have checked the examination paper and affirm it is complete.

Student Signature: ______ Date: ______

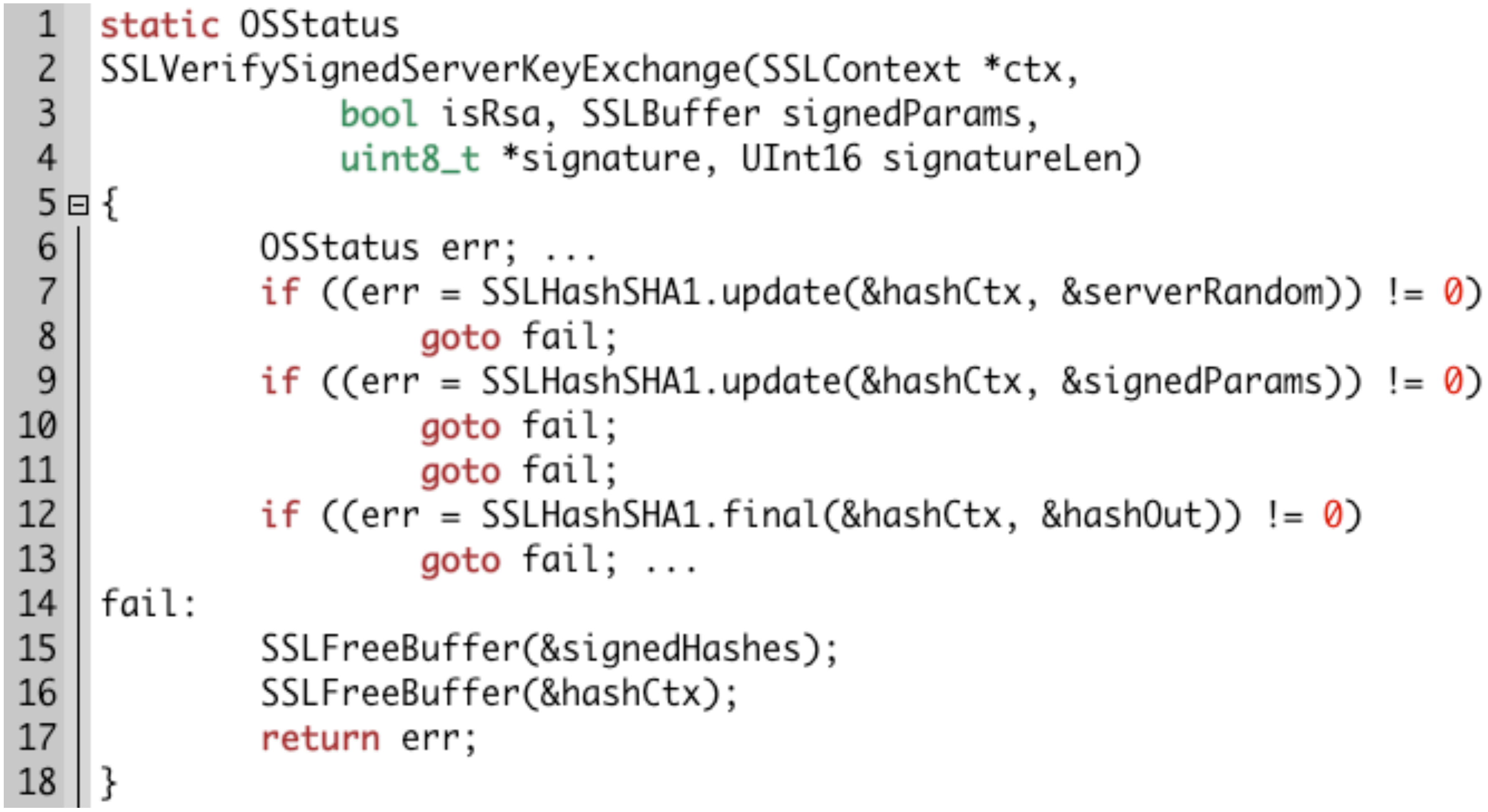
Office Use Only	
Question	**Mark**
Q 1-22	____ / 22
Q 23	____ / 8
Q 24	____ / 16
Q 25	____ / 18
Q 26	____ / 24
Q 27	____ / 12
Total:	____ **/ 100**

QUESTIONS

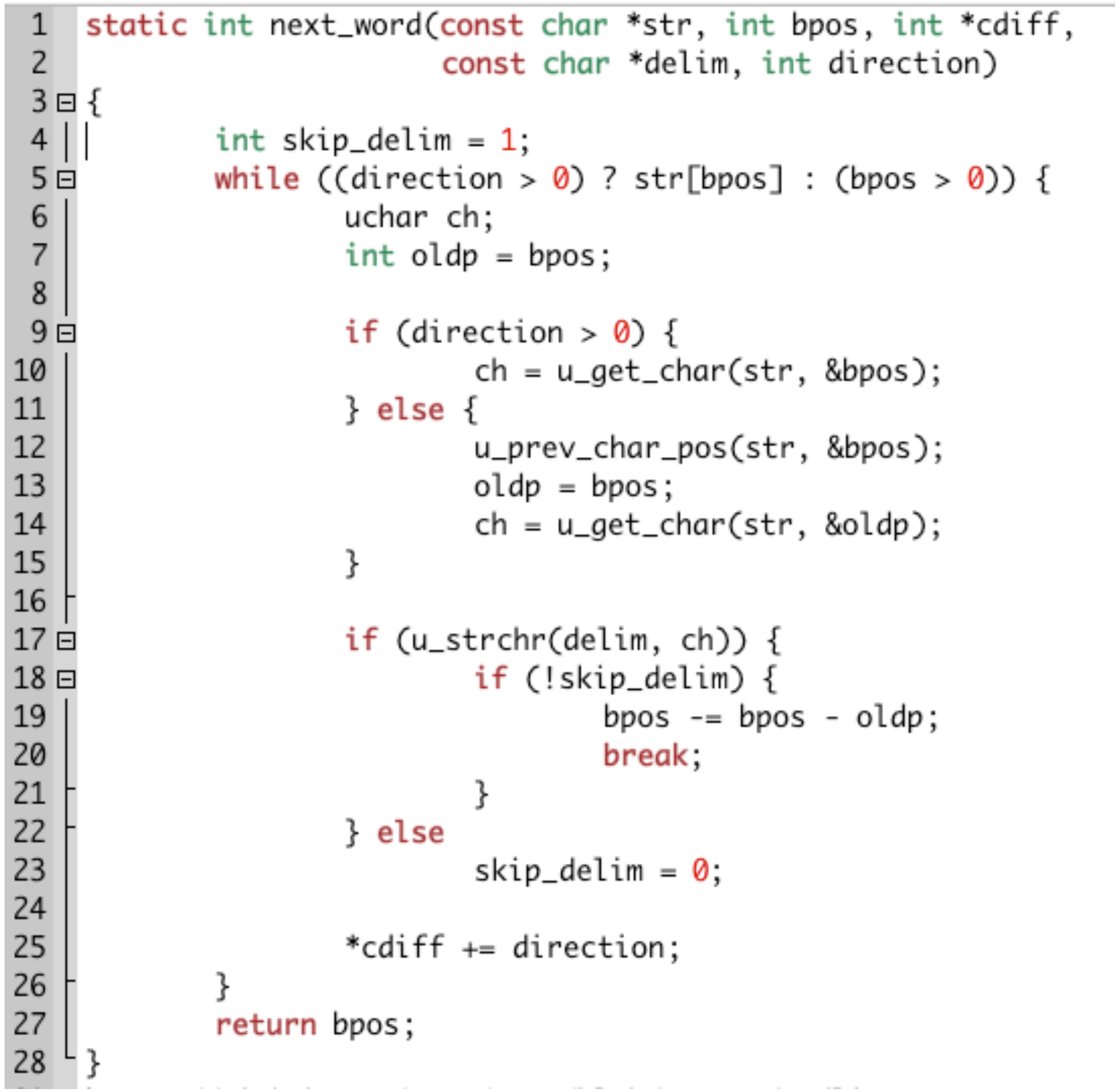
- 1-22 ANYTHING
- 23 – MEMORY
- 24 – PROGRAMMING
- 25 – MEMORY
- 26 – PROGRAMMING
- 27 – PROGRAMMING

- PROGRAMMING
  - Write a code solution for the given problem description

Quick review of sample C code



```c
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx,
            bool isRsa, SSLBuffer signedParams,
            uint8_t *signature, UInt16 signatureLen)
{
        OSStatus err; ...
        if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
                goto fail;
        if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
                goto fail;
                goto fail;
        if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
                goto fail; ...
fail:
        SSLFreeBuffer(&signedHashes);
        SSLFreeBuffer(&hashCtx);
        return err;
}
```



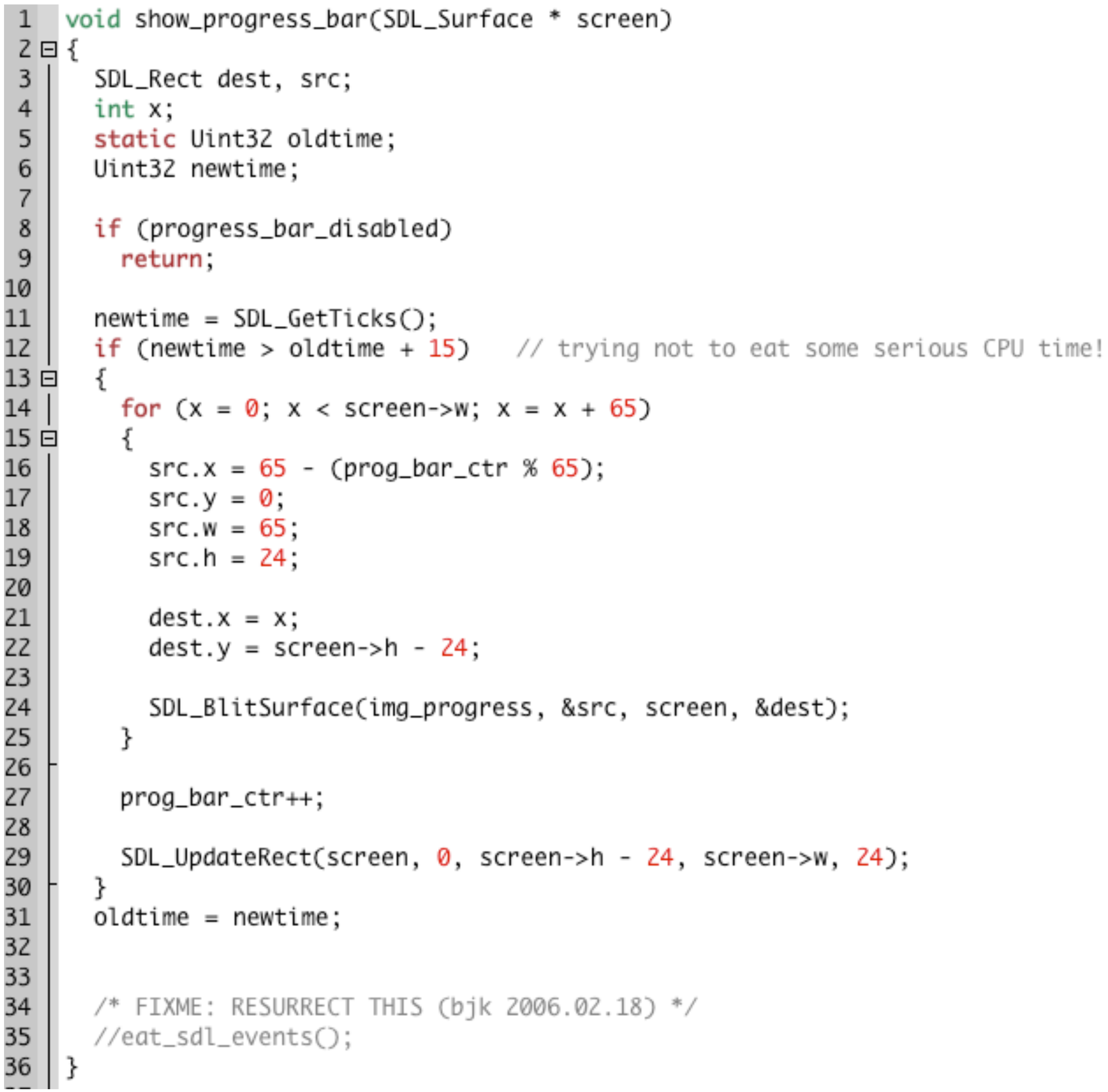
```c
static int next_word(const char *str, int bpos, int *cdiff,
                     const char *delim, int direction)
{
        int skip_delim = 1;
        while ((direction > 0) ? str[bpos] : (bpos > 0)) {
                uchar ch;
                int oldp = bpos;

                if (direction > 0) {
                        ch = u_get_char(str, &bpos);
                } else {
                        u_prev_char_pos(str, &bpos);
                        oldp = bpos;
                        ch = u_get_char(str, &oldp);
                }

                if (u_strchr(delim, ch)) {
                        if (!skip_delim) {
                                bpos -= bpos - oldp;
                                break;
                        }
                } else
                        skip_delim = 0;

                *cdiff += direction;
        }
        return bpos;
}
```

https://github.com/cmus/cmus/blob/master/cmdline.c



```c
void show_progress_bar(SDL_Surface * screen)
{
  SDL_Rect dest, src;
  int x;
  static Uint32 oldtime;
  Uint32 newtime;

  if (progress_bar_disabled)
    return;

  newtime = SDL_GetTicks();
  if (newtime > oldtime + 15)   // trying not to eat some serious CPU time!
  {
    for (x = 0; x < screen->w; x = x + 65)
    {
      src.x = 65 - (prog_bar_ctr % 65);
      src.y = 0;
      src.w = 65;
      src.h = 24;

      dest.x = x;
      dest.y = screen->h - 24;

      SDL_BlitSurface(img_progress, &src, screen, &dest);
    }

    prog_bar_ctr++;

    SDL_UpdateRect(screen, 0, screen->h - 24, screen->w, 24);
  }
  oldtime = newtime;


  /* FIXME: RESURRECT THIS (bjk 2006.02.18) */
  //eat_sdl_events();
}
```

http://tuxpaint.cvs.sourceforge.net/viewvc/tuxpaint/tuxpaint/src/progressbar.c?view=markup

Question 34 (Memory) (8 marks)

Read the following code segment and consider the memory contents after execution of all the statements up to line 10.

[redacted]

Each of the following boxes are labelled with their respective area of memory. Write the names of each variable and any data associated with them in the appropriate box.
`sizeof(char)` is 1 byte and `sizeof(void*)` is 4 bytes. The ASCII value of `'A'` is 65. Specify the starting address of each segment of memory and include the label for the data in your answer.

EXAMPLE CODE:

```
char a = 5;
char *b;
```

EXAMPLE MEMORY

0x0000 a 5
0x0001 b ?

Sample exam questions

Questions?

Where this fits

Java – Object Oriented

C - procedural

C++ Object Oriented

What subjects can you do after this subject?

2nd year

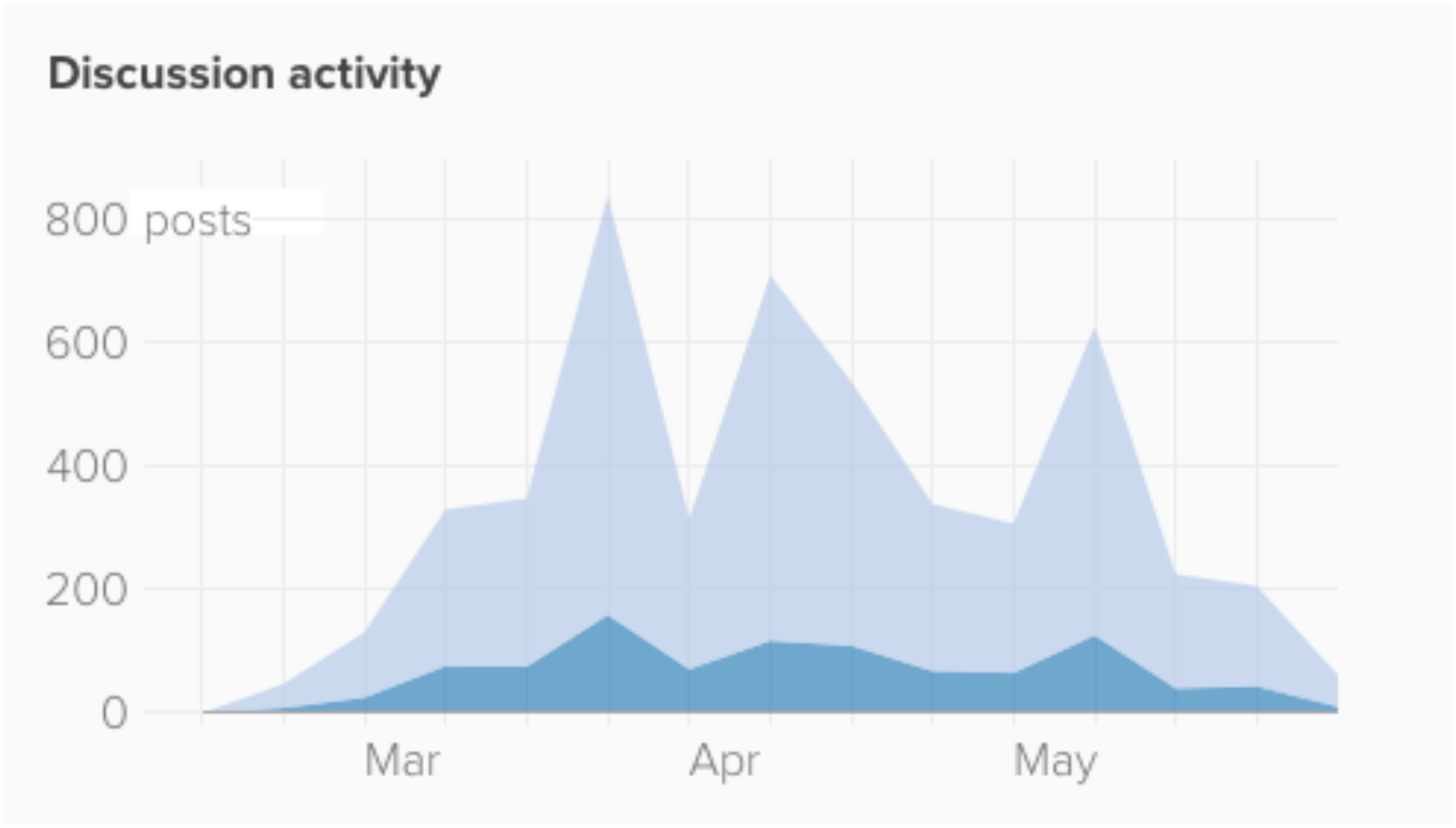
COMP2007 Algorithms and Complexity	COMP2121 Distributed Systems and Network Principles	INFO2110 Systems Analysis and Modelling	INFO2315 Introduction to IT Security

3rd Year

COMP3308 Introduction to Artifical Intelligence	COMP3419 Graphics and Multimedia	COMP3520 Operating Systems Internals	INFO3220 Object Oriented Design
COMP3109 Programming Languages and Paradigms	INFO3404 Database Systems 2	INFO3315 Human-Computer Interaction	COMP3456 Computational Methods for Life Sciences
COMP3530 Discrete Optimization	3rd Year projects COMP3615 ISYS3400 INFO3600		

C++

Need COMP2007


Discussion activity
800 posts
600
400
200
0
Mar
Apr
May

Credits

- Many people contributed to the preparation and delivery of the COMP2129 course.

- Lecture preparation
  - Bernhard Scholz
  - Tony Greening
  - Judy Kay
  - Bob Kummerfeld
  - John Stavrakakis (course coordinator)

The tutors!

- They help prepare all the lab sessions, quizzes and assignments. Answering so many questions on ed

- Teaching Assistant
  - Scott Maxwell
- Tutors
  - Thomas Chaffey
  - Harrison Rodgers
  - Patrick Nappa
  - Roger Lo
  - Tim Patten
  - Tyson Thomas
  - Justin Ting
  - Hisham Husain
  - Daniel Collis

Thank you

Good luck

https://www.youtube.com/watch?v=PS76m6ApOus

https://vimeo.com/127891298

http://www.nvidia.com/object/nvision08_gpu_v_cpu.html