MONASH University

MONASH
INFORMATION
TECHNOLOGY

UNIT REVIEW

LECTURE 11a / FIT2100 / SEMESTER 2 2019

WEEK 12 PART A

FINAL EXAM

CLOSED-BOOK WRITTEN EXAMINATION

EXAM DURATION

- 2 hours of writing time
- 10 minutes of reading time

- No calculators or reading materials permitted

EXAM FORMAT

- Total marks: 110
- Part A: Multiple choice (75 marks) – 25 questions
  - Same two-answer format as mid-sem test
- Part B: Short-answer questions (35 marks) – 3 multi-part questions
  - Similar to tutorial tasks. No code-writing.

PRE-EXAM CONSULTATIONS

HELD DURING SWOT-VAC WEEK

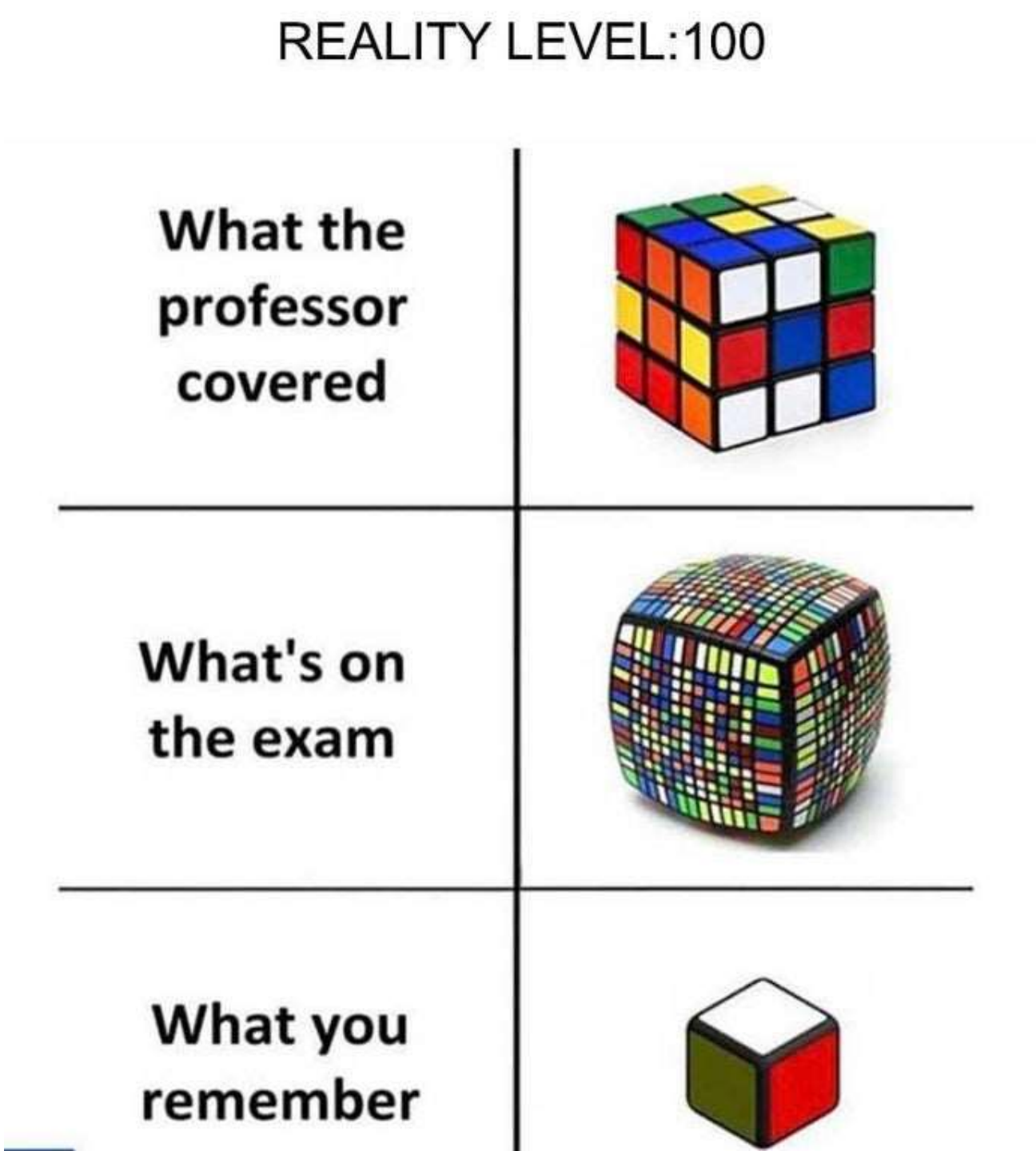
- Refer to dates and times posted on Moodle.

MEME

FUNNY (LAUGH)

- Edgy
- Relatable
- Cool

Aim for ***understanding*** *rather than just remembering points.*



MONASH University

GUIDE TO LECTURE WEEKS

THE FOLLOWING SLIDES SHOW A WEEK-BY-WEEK BREAKDOWN OF SUGGESTED POINTS FOR STUDY

- These learning points may help you focus your study on important points covered in lectures
  - This is not an exhaustive list, nor is it a list of exam questions
  - All materials from weeks 1 to 12 are examinable unless otherwise stated.
- For each week's lecture, refer also to the corresponding tutorial/lab (usually in the following week) for additional material.
- Also attempt revision questions posted under the **Exam** page on Moodle.

WEEK 1

COMPUTER SYSTEM OVERVIEW

- Understand the roles of different components of a computer system, and how they relate to each other
  - CPU
  - Main memory
  - I/O devices
    - Secondary storage
  - System bus

WEEK 2

OPERATING SYSTEM OVERVIEW

- Understand the role of the operating system kernel
- Understand the role of a system call in relation to program execution
- Understand the difference between ‘kernel’ and ‘user’ modes of execution as provided by the CPU
- The benefit of multiprogramming as a way of improving system utilization
- Understand what is meant by an ‘execution context’

WEEK 3

I/O MANAGEMENT AND HARD DRIVES

- Be able to compare and contrast characteristics of the three fundamental I/O techniques
  - Programmed I/O
  - Interrupt-driven I/O
  - Direct Memory Access (DMA)
- Be able to identify the difference between blocking and non-blocking I/O
- Understand the distinction between a block-oriented vs. a stream-oriented device.
- Identify the types of I/O buffers that are maintained within the OS itself

- Identify the arrangement of data on a hard drive (tracks, sectors, blocks/clusters, etc.)
- Understand the characteristics of the different disk-scheduling algorithms.

WEEK 4

FILE SYSTEMS

- Understand what is meant by fragmentation in the world of filesystems
- Understand how a FAT-based filesystem is laid out
- For inode-based filesystems, understand what information is stored in an inode
    - Understand how the blocks in a file are indexed within an inode
    - e.g. direct inode entries, single indirect inode entries, double-indirect, etc.
- Understand the purpose of the boot sector.

WEEK 5

PROCESSES

- Understand the purpose of a process control block (PCB)
  - What information is stored within a PCB
  - What information is stored outside a PCB
- Understand what transitions between process states can occur and why
  - Identify the different states and transitions in the five-state process model
  - Understand the reasons why a process may change state
- Understand what is meant by the ‘process image’ and the relationship to the process table, as well as the role of the process identifier.
- Have an understanding of what steps may be involved in switching between processes.

WEEK 6

UNIPROCESSOR SCHEDULING

- Be able to describe the different roles of short-, medium- and long-term scheduling.
- Understand why it might be useful for long-term scheduling to limit the degree of multiprogramming.
- Understand the meaning of the terms: turnaround time, throughput, response time.
    - Be able to discuss which of these measurements would be useful in different situations.
- Understand and be able to define the behaviour of the main process scheduling algorithms
    - Especially FCFS, RR, SPN, SRT.
- Be able to compare the different characteristics of preemptive vs non-preemptive systems.

WEEK 7

MID-SEMESTER TEST

NO LECTURES THIS WEEK

- Look back through the practice test.
- Make sure you are familiar with the two-answer system for multiple-choice questions.
    - The exam will have the same format.

WEEK 8

THREADS, CONCURRENCY PART 1

- Understand the differences between threads and processes.
  - What does a thread contain? What does a process contain? Why?
- Be able to compare advantages and disadvantages of using multiple threads vs. using multiple processes.
- Understand the difference between kernel-level threads and user-level threads.
- Be able to explain why concurrency issues are important when dealing with multiple processes or threads.
- Understand what a ‘race condition’ is and how mutual exclusion can prevent race conditions from happening.

WEEK 9

CONCURRENCY PART 2

- Understand what a semaphore does (binary and general/counting semaphores)
- Understand what a mutex does
    - and the difference between a semaphore and a mutex
- Remember and understand the four conditions needed for deadlock
- Understand what deadlock **prevention** means, and how one or more of the deadlock conditions can be removed from the system
- Understand what deadlock **avoidance** means, and how to apply the banker's algorithm to determine if a set of resource allocations leads to a deadlock (unsafe state).
    - Practice writing down the steps involved in following banker's algorithm. Remember that if a process can be chosen to proceed, it runs to completion and frees all its resources.
    - **Example step:** *P1 runs to completion and releases resources (4,2,0) to available vector.*
- Understand what deadlock **detection/recovery** means, and how a system might recover after deadlock.

WEEK 10

VIRTUAL MEMORY

- Understand how a logical address is translated into a physical address.
  - See also tutorial #7 (week 11)
- Understand the differences between paged and segmented systems.
  - And how fragmentation applies to main memory under each system
- Know what a page fault is.
- Know what a segmentation fault is.
- Understand how to carry out the steps of the most common page replacement algorithms
  - FIFO, LRU, OPT
  - …and the advantages/disadvantages of implementing those algorithms on a real system.
- How does shared memory relate to virtual memory?

WEEK 11

INTERPROCESS COMMUNICATION

- Be able to compare the characteristics of each IPC mechanism discussed.
  - How data is transmitted
  - If and how the data is structured
  - Whether the data is buffered and where
  - The concurrency implications of different IPC mechanisms (e.g. does the kernel guarantee FIFO behaviour when a sender and receiver share the same resource?)

WEEK 12

SECURITY

- Identify the categories of security threats on a system
- Understand how the operating system provides protection against possible security threats.
  - Through the filesystem
  - Through the virtual memory system
- Beware of **DARK** patterns.