You should attempt as many questions as you can before the examples class. You can discuss the questions with other students, but remember that the exam may contain similar questions, so make sure you can answer them by yourself.

Make sure that you can answer all the exercises given in the lectures

1. What features do “3-address” and “Load-Store” instruction sets have in common? How do they differ?

2. What features do the Load and Store instructions on the ARM have in common? How do they differ?

3. Without using a calculator, rewrite each of these (positive) numbers as (i) binary, (ii) octal, (iii) decimal, and (iv) hexadecimal:
   (a) 01011_2
   (b) 01101010_2
   (c) 22_8
   (d) 72_10
   (e) F_{16}
   (f) 7C_{16}

   Explain this “joke”:
   Q: Why do programmers always mix up Halloween and Christmas?
   A: Because Oct 31 is Dec 25.

4. How many different patterns can you make with:
   (a) 6 bits,
   (b) 9 bits,
   (c) 11 bits,
   (d) 15 bits.

5. How many bits would you need to count each of the following:
   (a) my 10 second-year tutees,
   (b) the 56 students taking COMP20121,
   (c) the 203 students taking COMP10031,
   (d) the 288 students taking COMP10900,
   (e) the 10^{57} atoms in the sun.

6. A memory contains 64k words each of 32 bits (where “k” is intended to mean 2^{10} i.e. 1024)
   (a) How many bytes is this equivalent to?
   (b) How many bits is this equivalent to?

   How many bits are needed to give a different pattern to locate each:
   (c) word
   (d) byte
   (e) bit

7. Computer words used to be 16 bits. Why did word size become 32 bits, rather than e.g. 24 bits. Why is the next size up 64 bits, rather than e.g. 40 bits?

8. What is the difference between initialising a variable when a program is loaded (before being run), and storing a value in a variable while the program is running? Write short pieces of ARM code to do these two things.