1. Exercises 1

2. What does a compiler do (see page 14)?
3. What does an assembler do (see page 14)
4. What is an assembly language (see page 14)
5. What is a machine language (see page 14)
6. Give example of high-level language (see page 15)
7. What is an integrated circuit (or chip) (see page 19)
8. What is a central processor unit (see page 19)
9. What does the data path do? (see page 19)
10. What does the control path do? (see page 19)
11. What is the interface between hardware and the lowest level of software known as? (see page 22)
12. Give an example of a volatile memory? (see page 22)
13. Is main memory the same as primary memory) (see page 23)
14. Can the secondary memory be a hard disk? (see page 23)
15. If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds, how much faster is A than B? (see page 31)
16. If the time for a complete clock cycle (e.g., 250 picoseconds, or 250 ps) and as the clock rate (see page 33)
17. Our favorite program runs in 10 seconds on computer A, which has a 2 GHz clock. We are trying to help a computer designer build a computer, B, which will run this program in 6 seconds. The designer has determined that a substantial increase in the clock rate is possible, but this increase will affect the rest of the CPU design, causing computer B to require 1.2 times as many clock cycles as computer A for this program. What clock rate should we tell the designer to target? (see page 34)
18. Suppose we have two implementations of the same instruction set architecture. Computer A has a clock cycle time of 250 ps and a CPI of 2.0 for some program, and computer B has a clock cycle time of 500 ps and a CPI of 1.2 for the same program. Which computer is faster for this program and by how much? (see page 35)
19. A given application written in Java runs 15 seconds on a desktop processor. A new Java compiler is released that requires only 0.6 as many instructions as the old compiler. Unfortunately, it increases the CPI by 1.1. How fast can we expect the application to run using this new compiler? (see page 40)
20. Which of computer A or B has highest MIPS when given is:
    a. computer A: instruction count 10 billion, clock rate 4GHz, and CPI 1.0
    b. computer B: instruction count 8 billion, clock rate 4GHz, and CPI 1.1?
    c. (see page 51)
21. A somewhat complex statement contains the five variables f, g, h, i, and j:
    a. f = (g + h) – (i + j);
    b. What might a C compiler produce? (see page 65)
22. What is a data transfer instruction? (see page 68)
23. Why is an address needed for a data transfer instruction? (see page 68)
24. What is the difference between a LOAD and a STORE instruction? (see page 68)

25. If a 32-bit integer: 0A0B0C0D (hexadecimal values) is stored in a memory at address as follows:
    a. address n: 0A
       address n+1: 0B
       address n+2: 0C
       address n+3: 0D
    b. is the organisation Big-Endian or Little-Endian?
    c. (see page A-43 or elsewhere for a clear explanation)
26. What is 123 (base 10) in a binary format? (see page 73)
27. What is the decimal value of this 64-bit two's complement number:
    a. $1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1000_{two}$
    b. (see page 79)
28. Convert the following hexadecimal into binary numbers:
    a. $eca8\ 6420_{hex}$
    b. (see page 81)
29. Convert the following binary number into hexadecimal:
    $0001\ 0011\ 0101\ 0111\ 1001\ 1011\ 1101\ 1111_{two}$
    a. (see page 81)
30. What is the stored-program concept (see page 86)
31. if register $s0 contained:
    a. $0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 1001_{two} = 9_{ten}$
    b. and the instruction to shift left by 4 was executed, what would the new value be?
    c. (see page 87)
32. For example, if register $t2 contains
    a. $0000\ 0000\ 0000\ 0000\ 0000\ 1101\ 1100\ 0000_{two}$
    b. and register $t1 contains
    c. $0000\ 0000\ 0000\ 0000\ 0011\ 1100\ 0000\ 0000_{two}$
    d. then, after executing the MIPS instruction
    e. and $t0,$t1,$t2 # reg $t0 = reg $t1 & reg $t2
    f. the value of register $t0 would be what? (see page 88)
33. What is the caller? (see page 98)
34. What is the callee? (see page 98)
35. What is the program counter? (see page 98)
36. What is a stack? (see page 98)
37. What is a stack pointer? (see page 98)
38. What does a push do? (see page 98)
39. What does a pop do? (see page 98)
40. What is PC-relative addressing (see page 115)?
41. Where do you find the operand in Immediate addressing (see page 117)?
42. Where do you find the operand in Register addressing (see page 117)?
43. In which order are compiler, assembler, linker and loader normally used? (see page 124)
44. What is an executable file (see page 126)?