Chapter 12 - Exercises

12.1 Opcodes, source operand reference, result operand reference and next instruction reference.

12.2 Main or virtual memory, processor register, immediate (contained in a field in the instruction being executed), and an I/O device.

12.3 The purpose could be to access or store information in memory or registers, or to hold data from arithmetic operations.

12.4 Operation repertoire – How many and which operations to provide.  
Data types: various types of data upon which operations are performed.  
Instruction format: Instruction length(in bits), number of addresses, size of various fields, ....  
Registers: Number of processor registers that can be referenced by instructions, and their use.  
Addressing: mode or modes by which address of an operand is specified

12.5 Addresses, numbers, characters, and logical data.

12.6 The last four bits in IRA bit pattern 011XXXX, the numbers 0-9 are represented using the same 4-bits as packed decimal.

12.7 The bits of a word are shifted left or right in a logical shift. The bit that is pushed out is lost on one end, while a zero is inserted on the other. The data is treated as a signed integer using arithmetic shift, and the sign bit is not shifted. A right shift replicates the sign bit into the bit position to its right, whereas a left shift performs a logic shift on all bits save the sign bit.

12.8 They are required for looping (allowing for repeated calls to instructions), allow for decision making, and it allows for breaking up programs into smaller pieces for easier programming.

12.9 One way is to test for some condition caused by an arithmetic operation, and checking for 0, positive, negative, or overflow. A three-address instruction format can also be used, with two addresses used for condition testing.

12.10 Being able to call a procedure from within a procedure.

12.11 Register, start of called procedure, or the top of the stack.