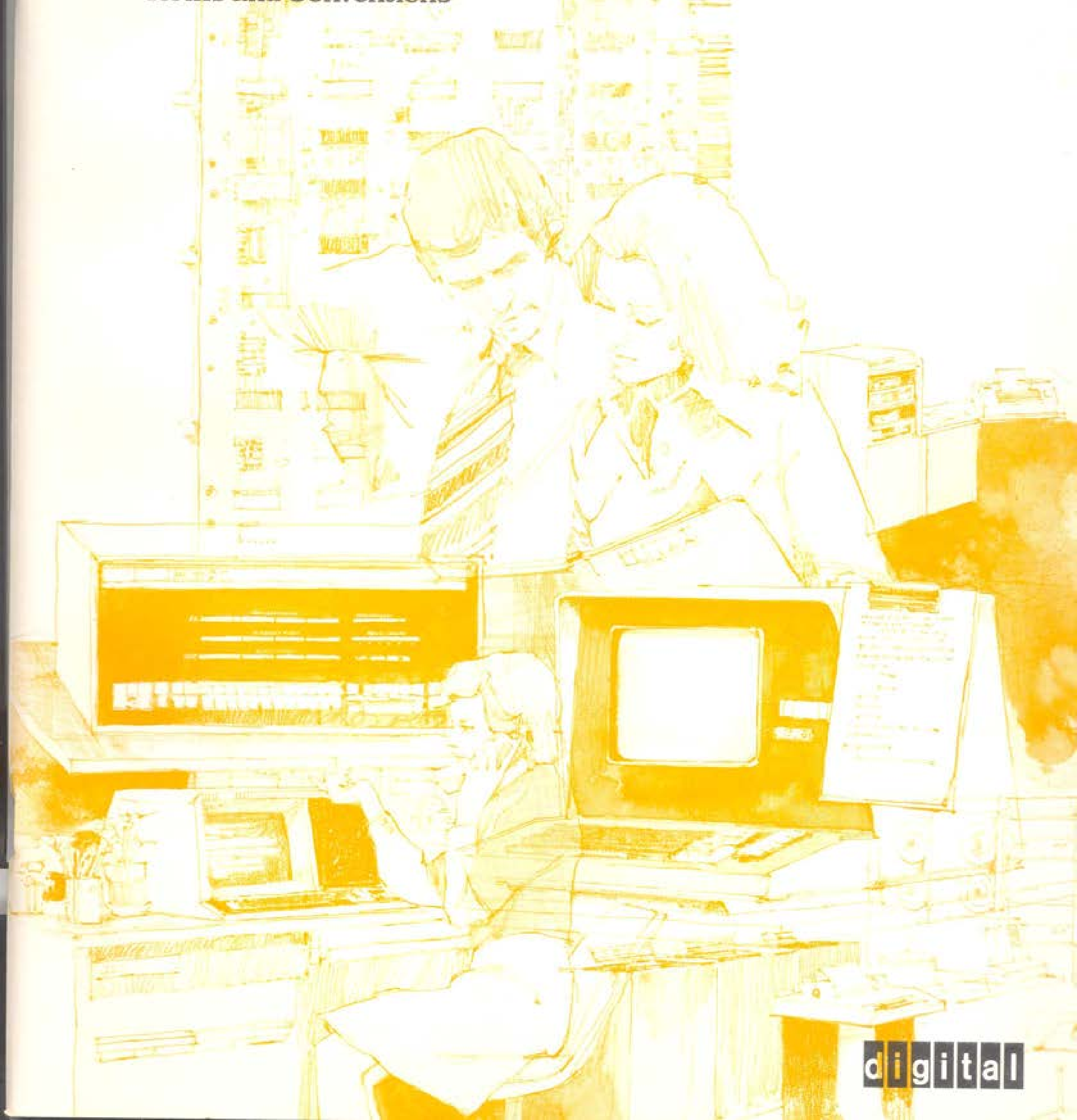
Introduction to Minicomputers

Terms and Conventions



digital

1st Printing, June 1976
2nd Printing (Rev), October 1977
3rd Printing, August 1979

Printed in U.S.A.

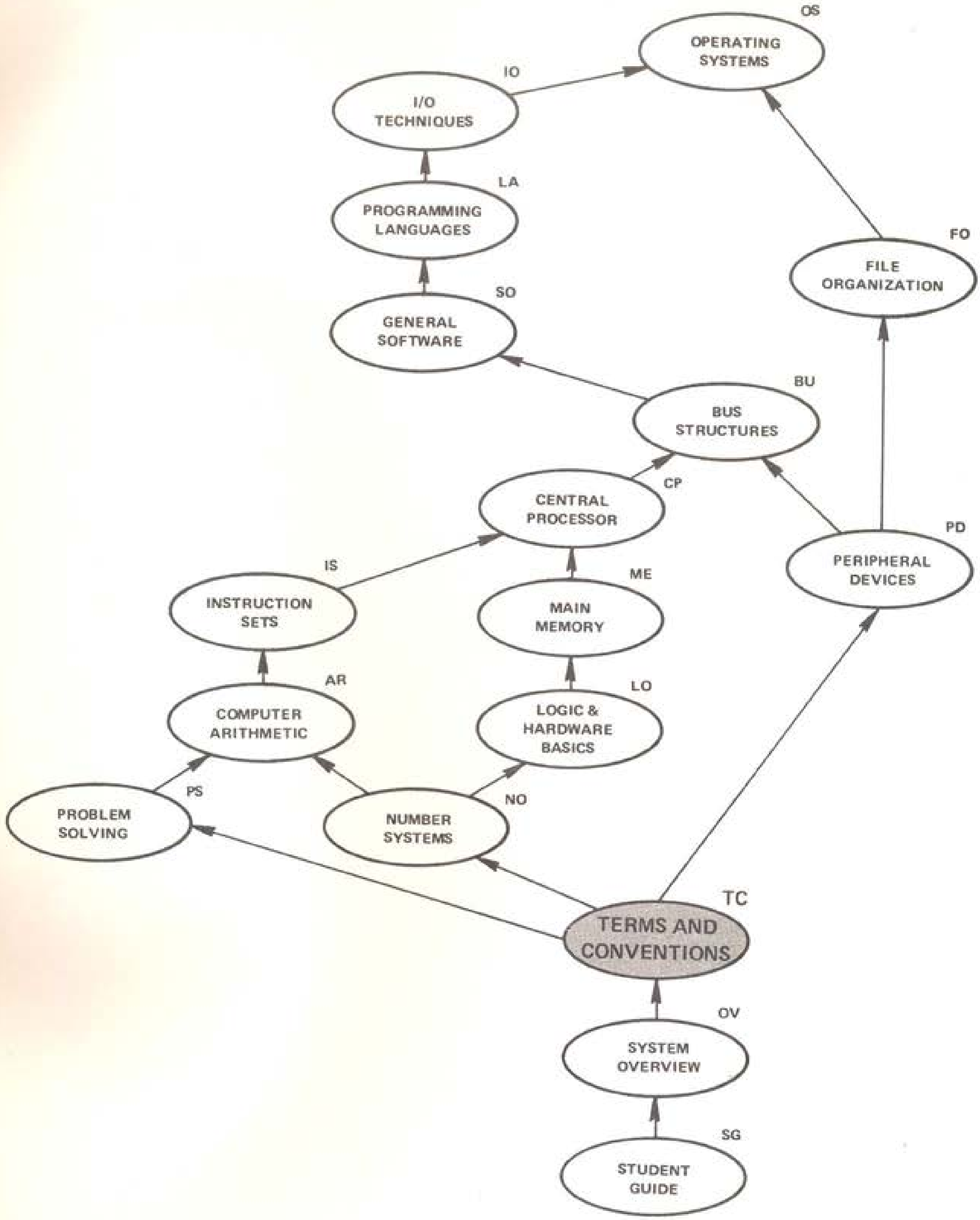
INTRODUCTION TO MINICOMPUTERS

Terms and Conventions

Student Workbook

Audio-Visual Course by Digital Equipment Corporation

COURSE MAP


OS
OPERATING SYSTEMS
IO
I/O TECHNIQUES
LA
PROGRAMMING LANGUAGES
FO
FILE ORGANIZATION
SO
GENERAL SOFTWARE
BU
BUS STRUCTURES
CP
CENTRAL PROCESSOR
PD
PERIPHERAL DEVICES
IS
INSTRUCTION SETS
ME
MAIN MEMORY
AR
COMPUTER ARITHMETIC
LO
LOGIC & HARDWARE BASICS
PS
PROBLEM SOLVING
NO
NUMBER SYSTEMS
TC
TERMS AND CONVENTIONS
OV
SYSTEM OVERVIEW
SG
STUDENT GUIDE

CONTENTS

Terms and Conventions

Introduction

People use computers to process information automatically at fantastic rates of speed. A computer can accomplish a *lifetime* of work in minutes or at most in a few hours. For example, a very primitive computer was able to perform a calculation in 20 minutes that would have taken 100 mathematicians 100 years to complete. Today's modern computers can perform *more than* one million computations accurately in one second.

Remember the computer *never* makes a mistake; it will do *exactly* what it has been programmed to do – no more, no less. It does not become tired or bored with repetitive operations and has never been known to take a coffee break.

If you understand certain basic terms and established conventions that are used with all digital computers, you will find it easier to comprehend how digital computers operate. Because of the extensive and growing use of computers, the vocabulary of computer-related terms and conventions has been expanding rapidly. Common words, such as "address" and "location," take on highly specialized meanings when used in reference to computer operations. Computer terms and conventions can be divided into three distinct categories and, therefore, are described in three separate lessons in this module.

The first lesson deals with *units of information*. All information processed by a digital computer, whether data or instructions, can be divided into units of various sizes ranging from the smallest element of information a computer can recognize to much larger elements that the computer typically works with. This lesson describes the forms of the various units of information used by digital computers as well as the terms used to describe these units.

The second lesson deals with *storage and retrieval of information.* Any information to be used by the computer must first be stored where the computer can find it when needed. When the computer wants some specific information, it must first locate it within the storage area (memory) and then retrieve it from the stored location. This lesson describes the storage arrangement of memory, the methods used by the computer to transfer information in and out of memory, and the terms related to storage and retrieval operations.

The third lesson deals with *units of time.* Because digital computers operate at such high speeds, a single operation can be performed in an extremely small fraction of a second. Thus, when describing the operating speeds of digital computers, units of time that pertain to fractions of a second are used. These various time units are described in this lesson. The lesson also includes an explanation of the standard mathematical symbols used throughout this course.

Units of Information

OBJECTIVES

1. Given the names of the three basic units of computer information, be able to list them in order according to size, beginning with the smallest unit.

2. Given four possible sizes of a byte, be able to select the one or ones specifying a byte size.

SAMPLE TEST ITEMS

1. The three basic units of computer information are listed below. Write an A next to the smallest unit, a B next to the next larger unit, and a C next to the largest unit of computer information.

Basic Unit	Size
Word	____
Bit	____
Byte	____

2. Circle the letter of the answer that specifies the correct size of a *byte*.

   a. One-half of a word

   b. One-quarter of a word

   c. 6 bits long

   d. 8 bits long

   e. All of the above

   f. Two of the above

Mark your place in this workbook and view Lesson 1 in the A/V program, "Terms and Conventions."

A digital computer cannot work *directly* with decimal numbers, letters of the alphabet, or special symbols (such as ?, $, %, etc.). Although the computer may consist of thousands of electronic components, each basic component has only two operating levels: *on* and *off*. Because each component has only two possible operating levels, it can only represent two values: 0 and 1. When any component is electronically switched to the *on* level, it represents the value 1. On the other hand, when it is electronically switched to the *off* level, it represents the value 0. Thus, whenever a digital computer is operating, each one of its thousands of internal components must represent either a 0 or a 1. In other words, a computer can only process 0s and 1s. Because of this, all computer information, whether instructions or data, must be converted to a specific combination of 0s and 1s in order for the computer to be able to recognize and handle the information.

Figure 1 shows how information is represented within a computer by a specific combination of 0s and 1s. In this example, the alphabetic character "M" is represented by the states of seven individual electronic circuits. Notice that each circuit is in one of two levels. That is, each circuit is either *on* or *off*, thereby representing either a 1 or a 0. The particular combination shown in the figure (1 0 0 1 1 0 1) is a representation of the letter "M" that the computer can recognize.

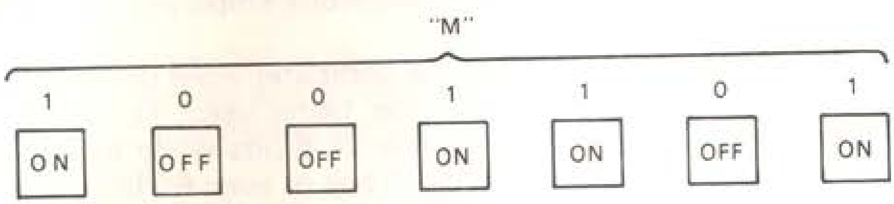


Figure 1 Representing Information

Bit

Because a computer uses only *two* digits (0 and 1), they are referred to as *binary* digits. (The prefix *bi* indicates two. For example, a *bi*cycle is a 2-wheeled cycle as opposed to a tricycle.) Rather than use the term "binary digit," people who use computers prefer the term "bit." "Bit" is an abbreviation formed by taking the first two letters of "binary" and combining them with the last letter of "digit" (*BI*nary digi*T*). Thus, a binary digit, or "bit," is the *smallest* unit of information used in digital computers. A bit must always be either a 0 or a 1.

Bits are usually awkward to work with. In addition, computers can *not* store or retrieve individual bits when transferring information to or from memory. Therefore, the computer works with larger elements of information called "bytes" and "words." Both a byte and a word contain a number of bits.

Word

Bits are grouped together and handled as a single entity known as a word. A computer word consists of a *fixed number* of bits that are processed as a *single unit* of information. Digital computers are word-oriented; that is, they are designed to *store* individual *words* in memory and to *retrieve* individual *words* from memory.

A *single* computer word might represent a single decimal number, or three decimal numbers, or two letters of the alphabet, or a few special symbols, or perhaps an actual English word. In other words, the significance of a computer word depends on the computer being used, the size of the word, and the specific conventions employed by the user.

The number of bits contained in one computer word depends on the design and size of the digital computer being used. For instance, a minicomputer word may contain as few as 8 bits while a large-scale computer might use a word length of 36 bits or even 60 bits. The most common word lengths used by minicomputers are 12, 16, and 18 bits per word.

Because information used by a computer may be either *instructions* or *data,* a computer word may represent either instructions or data. If the word contains *data* to be processed, it is called a *"data word."* If the word contains an *instruction,* it is called an *"instruction word."*

Byte

In *some* digital computers, words are divided into smaller subunits called *"bytes."* A byte is usually either one-half or one-quarter the size of the computer word. Bytes, like words, contain a *fixed number* of bits and are treated as a *single unit* of information. Bytes are used because it is often easier to work with a unit smaller than a full word, especially if the computer word contains a large number of bits.

Because a byte is either one-half or one-quarter the word length, the number of bits in a byte depends on the size of the computer word. For example, if a minicomputer employs a 12-bit word, then the word could be divided into two 6-bit bytes. On the other hand, if a 16-bit word were used, then each byte would contain 8 bits. Because most bytes, by convention, are either 6 or 8 bits in length, they are used to represent quarter word lengths only when larger word sizes are used. For instance, a 32-bit word would normally be divided into *four* 8-bit bytes. On the other hand, an 8-bit *word* would not be subdivided into bytes at all.

Bytes are similar in function to words in that they may consist of either data or instructions and they may be stored or retrieved in memory. The byte is the smallest unit of information that can be addressed. Note that not all computer memories are byte-addressable.

Summary

As shown in Figure 2, all computer information, whether data or instructions, consists of *bits, bytes,* and *words.*

The *bit,* or binary digit, is the *smallest* unit of information that can be used by the computer and is always either a 0 or a 1. Bits are grouped together to form bytes and words which are handled as single entities by the computer.

A computer *word* can be addressed (that is, either stored in or retrieved from memory) while a bit cannot. Depending on the type of computer used, a word may consist of anywhere from 6 to 60 bits.

A *byte* is one-half or one-quarter the size of a computer word, and is typically either 6 or 8 bits long. The byte is the smallest unit of information that can be addressed. Not all computers are byte-addressable.

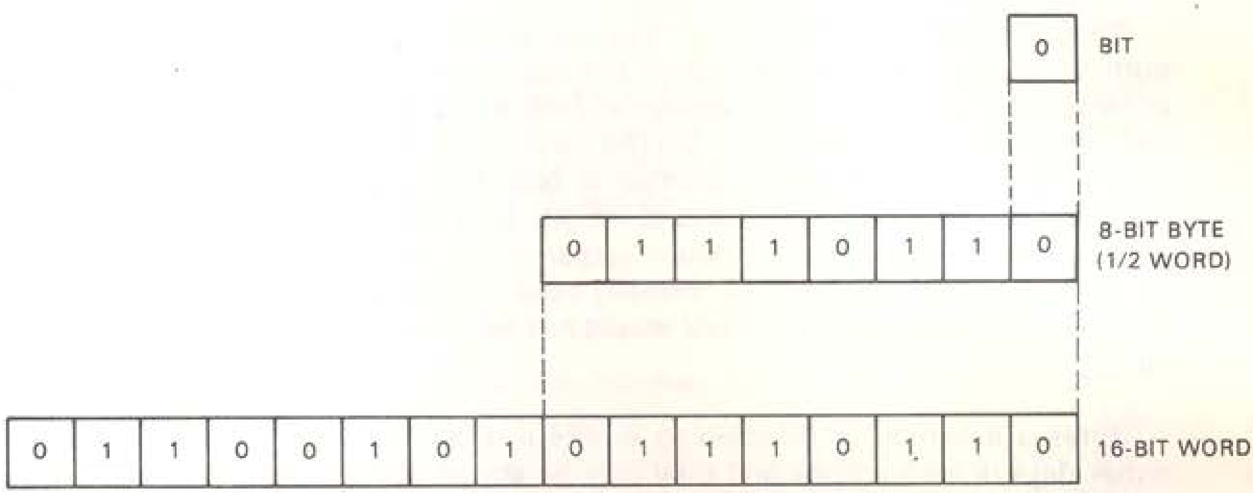


Figure 2 Relationship of Bit, Byte, and Word

Before going to the next lesson, do the practice exercises that begin on the following page. There are eight exercises.

EXERCISES

Fill in the blank or circle the correct answer.

1. The word "bit" is actually an abbreviation for ______________ .

2. A bit must always be either a __________ or a __________ .

3. The smallest unit of information that can be used by a computer is:

   a. A bit

   b. A byte

   c. A word

4. The smallest unit of information that can be stored in or retrieved from memory is:

   a. A bit

   b. A byte

   c. A word

5. In the space provided below, draw a 12-bit word. Then draw one byte of the word, and one bit of the word.

SOLUTIONS

1. The word "bit" is actually an abbreviation for *binary digit.*

2. A bit must always be either a *1* or a *0.*

3. The smallest unit of information that can be used by a computer is:

   (a.) A bit

   b. A byte

   c. A word

4. The smallest unit of information that can be stored in, or retrieved from, memory is:

   a. A bit

   (b.) A byte

   c. A word

5. In the space provided below, draw a 12-bit word. Then draw one byte of the word, and one bit of the word.

0	1	0	1	1	0	1	1	0	1	1	0	WORD (12 BITS)

1	1	0	1	1	0	BYTE (6 BITS)

0	BIT

EXERCISES

Circle the correct answers or fill in the blanks.

6. The size of a computer word depends on:

   a. The number of bytes

   b. The instruction size

   c. The computer design

   d. The task being performed

7. A byte may be:

   a. One-half of a word

   b. One-quarter of a word

   c. 6 bits long

   d. 8 bits long

   e. All of the above

   f. None of the above

8. Name the three basic units of computer information. Place them in order according to size, beginning with the largest unit.

   a. ____________

   b. ____________

   c. ____________

6. The size of a computer word depends on:

   a. The number of bytes

   b. The instruction size

   (c.) The computer design

   d. The task being performed

7. A byte may be:

   a. One-half of a word

   b. One-quarter of a word

   c. 6 bits long

   d. 8 bits long

   (e.) All of the above

   f. None of the above

8. Name the three basic units of computer information. Place them in order according to size, beginning with the largest unit.

   a. *Word*

   b. *Byte*

   c. *Bit*

Storage and Retrieval of Information

OBJECTIVES

1. Given the terms "memory address," "memory contents," and "memory location" and a list of three definitions, be able to match each term with its definition.

2. Given four sample memory locations having similar address and content notations but different formats, be able to select the one example that is constructed in the proper format.

3. Given sample memory sizes expressed in K notation, be able to convert each sample into its equivalent decimal notation.

SAMPLE TEST ITEMS

1. Match each memory term with its definition.

Memory Terms		Definitions
a. Memory address b. Memory contents c. Memory location	____	Used for storage of one word (or byte) of information.
	____	Changes when information is stored; does not change when information is retrieved.
	____	Specifies a particular place in memory that can never change.

2. Circle the letter of the one example that is the correct format for a single memory location with an address of 5000 and contents of 32.

a. [32] 5000   b. 5000 [32]

c. 32 [5000]   d. [5000] 32

3. The *exact* number of individual locations in a 16K memory is ____________.

Before a digital computer can perform any job, it must have two types of information: the *data* that is to be processed, and the *instructions* that tell the computer what type of processing to perform. These instructions and data are stored in main memory until the central processing unit is ready to take the information out of memory (retrieve it) for processing.

Because main memory can store thousands of pieces of information, the memory must be arranged in a fashion that will allow information to be stored in a specific spot so that the central processing unit will know where to go in order to obtain the information it needs to perform a particular job.

This lesson describes how main memory is arranged, how the central processing unit stores and retrieves information, and the various terms associated with these storage and retrieval operations.

Mark your place in this workbook and view Lesson 2 in the A/V program, "Terms and Conventions."

Memory Structure

A computer's main memory consists of thousands of individual components, or cells, which are referred to as memory "locations." Each location is used to store information that the CPU needs for processing, or to store the results of the processing. An individual memory location can hold only one piece of information at a time. That is, one location can store only a single instruction word or a single data word. (If a particular computer uses "bytes," then each location will hold only one *instruction byte* or one *data byte.*)

Every location in main memory is identified by a unique address that has been assigned to it at the time of manufacture. It is important to remember that this address specifies the *location, not the contents* of that location. It is similar to a house address which specifies the location of the house rather than identifying the names of the people who might be living there at the time. Memory addresses are numbers that usually begin at 0 and progress to the highest number required to identify all of the locations within a specific memory.

Because each location has its own unique address, the CPU can go *directly* to any location in memory and either store a word or retrieve a previously stored word.

Although the *address* of a location can never be changed, the *contents* of a location can be changed whenever desired. For instance, assume that the CPU is dealing only with the memory location that has an address of 0056. The CPU might retrieve an ADD instruction from location 0056, perform the necessary addition, and then store the answer of 32477 in location 0056. Now the location having an address of 0056 contains the *data* value 32477 rather than the ADD *instruction*.

Figure 3 shows a portion of main memory and indicates the relationship between a memory *location,* the *address* of that location, and the *contents* of that location.

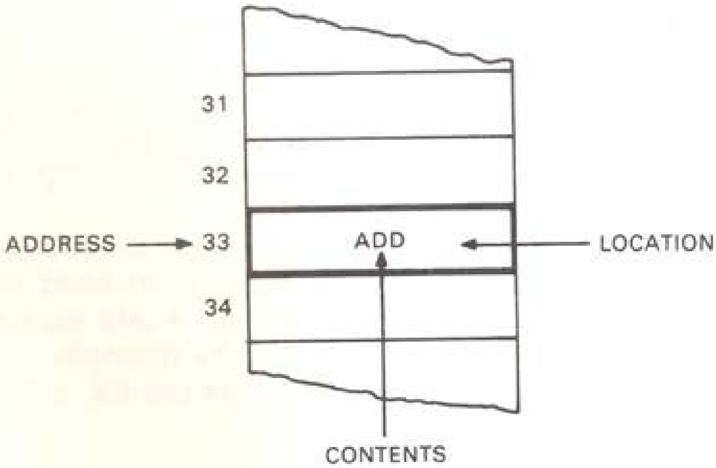


Figure 3 Segment of Main Memory

Storing and Retrieving Information

When any memory location is to be used for either storing or retrieving information, the CPU must first specify the *address* of the location that is to be used. This action is referred to as *addressing memory.*

Once a memory location has been addressed, information can be moved into the specified location for storage. Whenever new information is *stored* in a memory location, it replaces (or destroys) the previous *contents* of that location.

When the CPU wants to *retrieve* information from memory, it addresses the memory location and removes its contents. However, during retrieval operations, the *contents* of that location are *not destroyed.* Rather, a *copy* of the data is retained in the memory location. Thus, information stored in memory can be used over and over again.

Memory Size

Computer memories are available in many different sizes. Memory size refers to the *total number of words* that can be stored in a specific memory. Because one location can be used to store one word, memory size also refers to the *total number of locations* in memory.

The letter "K" is normally used to indicate the size of a particular memory. Used in this manner, "K" represents the value 2^{10}, which equals 1,024. Thus, a 1K memory contains 1,024 storage locations.

Although "K" represents 1,024, it is a common practice to round off this value to 1,000. Thus, "K" is basically a short-hand notation for 1,000. For instance, a computer memory with 4,096 locations would be referred to as a 4K memory. Memory size normally increases by multiples of 4K. Thus, common memory sizes are 4K, 8K, 12K, 16K, etc.

Although a 32K memory contains roughly 32,000 storage locations, it is necessary to use the actual value of "K" (1,024) to find the *true* number of locations. A 32K memory, for instance, actually contains 32,768 locations (32 times 1,024).

It is important to remember that each memory location is assigned a *unique address.* Thus, if a memory has 32K locations, there must be approximately 32,000 *individual* addresses. In actuality, a 32K memory has 32,768 locations and would, therefore, have that many individual addresses.

Before going on to the next lesson, do the practice exercises that begin on the following page. There are six exercises.

EXERCISES

1. Match the appropriate term with the proper definition by placing the letter associated with the term in the parentheses next to the definition. One term may apply to more than one definition.

a. Memory location
b. Memory address
c. Memory contents

( ) A number that specifies a particular location

( ) Can never change

( ) Can be changed when desired

( ) Used to store information

( ) Consists of either data or an instruction

( ) Must have a unique address

( ) Can hold only one word (or byte)

( ) Does not change when information is retrieved

( ) CPU can go directly to this

( ) Changes when information is stored

2. Draw a single memory location which has an address of 335 and is storing the value 73245. Label the location, the address, and the contents.

SOLUTIONS

1. Match the appropriate term with the proper definition by placing the letter associated with the term in the parentheses next to the definition. One term may apply to more than one definition.

a. Memory location
b. Memory address
c. Memory contents

(b) A number that specifies a particular location

(b) Can never change

(c) Can be changed when desired

(a) Used to store information

(c) Consists of either data or an instruction

(a) Must have a unique address

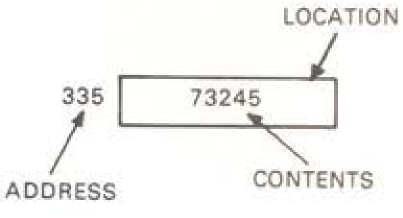
(a) Can hold only one word (or byte)

(c) Does not change when information is retrieved

(a) CPU can go directly to this

(c) Changes when information is stored

2. Draw a single memory location which has an address of 335 and is storing the value 73245. Label the location, the address, and the contents.



EXERCISES

3. The following questions pertain to the section of main memory drawn below:

	3477
201	2220
202	1123
	0000
204	5521
	26
206	354

a. What is the address of the location containing the value 3477? ____________

b. Which location (not address) has all zeros? ____________

c. If information was retrieved from location 205, what would the contents of that location be after the retrieval operation? ____________

d. If the value in location 201 were retrieved and then stored in location 202, what would be the contents of location 201? ____________

What would be the contents of location 202? ____________

3. The following questions pertain to the section of main memory drawn below:

	3477
201	2220
202	1123
	0000
204	5521
	26
206	354

a. What is the address of the location containing the value 3477? 200

b. Which location (not address) has all zeros? the fourth

c. If information was retrieved from location 205, what would the contents of that location be after the retrieval operation? 26 (retrieval does not change the contents)

d. If the value in location 201 were retrieved and then stored in location 202, what would be the contents of location 201? 2220

What would be the contents of location 202? 2220

NOTE

Retrieving information does not change the contents of a location; therefore, location 201 would still have the value 2220.

On the other hand, storing information destroys the previous value, so location 202 would now have the new value of 2220.

EXERCISES

4. A 64K memory has approximately __________ locations.

5. A 128K memory has approximately __________ locations.

6. Write the actual (true) number of locations that are available for each of the following memory sizes:

   a. 32K __________

   b. 16K __________

   c. 72K __________

   d. 8K __________

   e. 128K __________

   f. 4K __________

SOLUTIONS

4. A 64K memory has approximately <u>64,000</u> locations.

5. A 128K memory has approximately <u>128,000</u> locations.

6. Write the actual (true) number of locations that are available for each of the following memory sizes:

NOTE

To obtain the actual number of locations, it is necessary to use the value 1,024 for "K." Thus, multiplying the size of the memory by 1,024 will give the actual number of locations.

For instance, in d. below, 8K is equal to 8 times 1,024 or 8,192 locations.

a.	32K	32,768 locations
b.	16K	16,384 locations
c.	72K	73,728 locations
d.	8K	8,192 locations
e.	128K	131,072 locations
f.	4K	4,096 locations

Units of Time and Arithmetic Functions

OBJECTIVES

1. Given the terms "nanosecond," "millisecond," and "microsecond" and four pairs of abbreviations for each term, be able to select the correct pair of abbreviations for each term.

2. Given a specific unit of time expressed as nanoseconds, be able to convert it to its equivalent fraction of a second.

3. Given a list of five standard and conditional symbols and ten functions, be able to match each symbol with its function.

4. Given expressions and symbolic representations for each expression, be able to select the proper symbolic representation(s) for each expression.

SAMPLE TEST ITEMS

1. Circle the letter of the correct pair of abbreviations for each of these terms.

Nanosecond

a. nsec or nanosec

b. ns or nsec

c. ns or nanosec

d. nasec or nansec

Millisecond

a. ms or msec

b. ns or ksec

c. ksec or milsec

d. msec or milsec

Microsecond

a. msec or microsec

b. msec or ms

c. μs or microsec

d. μs or ksec

SAMPLE TEST ITEMS

2. The equivalent fraction of a second for 700 nanoseconds is _____________.

3. Five symbols and ten functions are listed below. Match each symbol with its function.

Symbol	Function
⩾	______
:	______
=	______
<	______
↑	______

a. Less than
b. Less than or equal to
c. Raise to the power of
d. Find the square
e. Multiply
f. Divide by
g. Greater than
h. Greater than or equal to
i. Compare
j. Equal to

4. Indicate which of the following symbols represents the expression stated by circling the correct letter.

X raised to the 12th power

a. X:12

b. X↑12

c. X**12

d. Both A and B

e. Both B and C

R times X

a. R**X

b. R:X

c. R*X

d. R (x) X

e. None of the above

A typical digital computer can perform one million *or more* additions in *one second.* Because a computer can perform so many calculations every second, individual operations are measured in minute fractions of a second. This lesson describes the units of time used to measure computer operations. This lesson also explains some of the standard symbols that will be used throughout this course.

> Mark your place in this workbook and view Lesson 3 in the A/V program, "Terms and Conventions."

Measuring Time

The most common units of time used in digital computers are: *milliseconds, microseconds,* and *nanoseconds.* These units are listed in Table 1 below.

Table 1 Units of Time

Units of Time	Fraction of a Second	Abbreviation
millisecond	thousandth (1/1000)	ms or msec
microsecond	millionth (1/1,000,000)	μs or microsec
nanosecond	billionth (1/1,000,000,000)	ns or nanosec

Many of the operations that occur in input units (such as a teletypewriter) or output units (such as a line printer) are measured in *milliseconds.* However, a millisecond is too long a time interval to accurately measure CPU or memory operations. Therefore, these operations are measured in *microseconds* or *nanoseconds.* For example, a computer might perform one complete addition in just 2.6 *microseconds* while the CPU can locate and retrieve a word from memory in just 800 *nanoseconds.*

Figure 4 is a graphic depiction of units of time that will help you understand the fractional units we have been discussing.

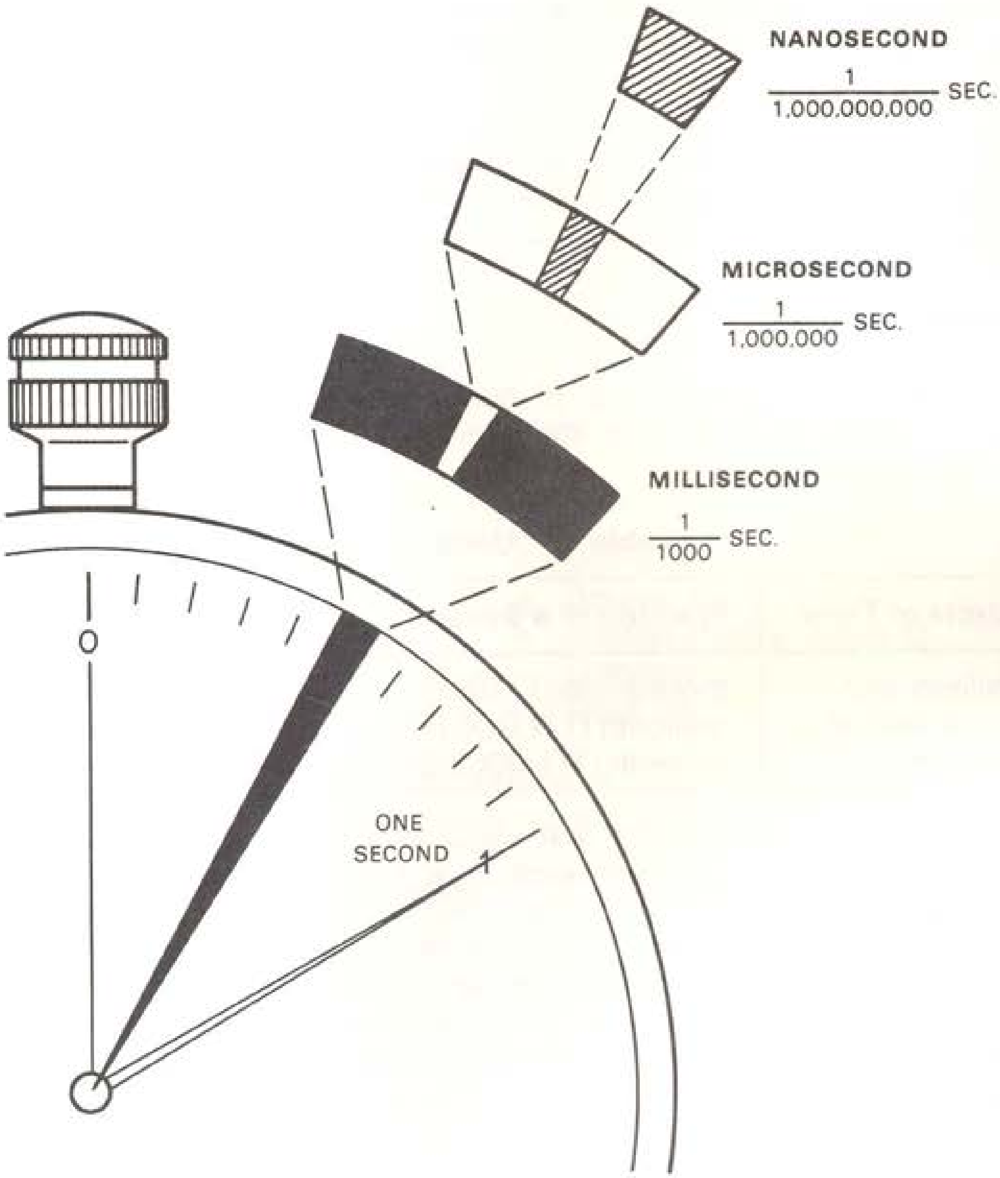


Figure 4 Units of Time

Standard Symbols

Table 2 lists commonly used arithmetic symbols that will occur throughout this course. The symbols for add, subtract, and divide are self-explanatory and are the same symbols used in normal arithmetic.

Note, however, that the symbol for the multiply operation is an asterisk (rather than the conventional arithmetic symbol X). When dealing with computers, the convention is to use the asterisk for the multiplication symbol.

Notice also that either one or two symbols may be used to show a number raised to a certain power. As shown in the table, 2 ↑ 4 indicates 2^4 while Y ** 7 indicates Y^7. Raising a number to a power is also referred to as *exponentiation* because it is identical to placing an *exponent* after the number. For instance, the expression 2^4 can be read as either "two raised to the fourth power" or as "two with an exponent of four." In either case, the function of the exponent (or power) is the same. It indicates how many times the base number is multiplied by itself. Thus, 2^4 indicates that the number 2 is multiplied by itself 4 times (2 times 2 times 2 times 2).

Table 2 Arithmetic Symbols

Symbol	Meaning	Example
+	add	Y + 7
–	subtract	Y – 7
*	multiply	Y * 7
/	divide	Y/7
↑ or **	raise to the power of	2 ↑ 4 Y ** 7

Conditional Symbols

Computers are often used to compare two quantities to determine if they are equal or not equal, or to determine if one quantity is larger or smaller than the other. The symbols used to express these conditions are listed in Table 3.

Table 3 Conditional Symbols

Symbol	Meaning	Example	Comments
:	compare	A : B	A is compared with B
=	equal to	A = B	A is equal to B
≠	not equal to	A ≠ B	A is not equal to B
>	greater than	X > 7	X is greater than 7
<	less than	X < 7	X is less than 7
⩽or <=	less than or equal to	X ⩽ 7 or X <= 7	X is less than or equal to 7
⩾or >=	greater than or equal to	X ⩾ 7 or X >= 7	X is greater than or equal to 7

The Symbol =

In most programming languages the = sign has two interpretations. In the previous section we discussed the = sign used as a *conditional* symbol: A = B. We were testing for an equality condition; that is, whether A precisely equals the value of B.

The second way in which we can use the = symbol is when we write equations for computer programs. In this context the meaning of the = sign is interpreted as "is replaced by." For example, the *equation,* A = B, is read as "the value of A *is replaced by* the current value of B."

> Now do the following exercises on units of time and standard symbols before taking the module test.

EXERCISES

1. Write the proper abbreviation for each of the following terms:

   a. microsecond ____________

   b. nanosecond ____________

   c. millisecond ____________

2. Write the equivalent fraction of a second for the following terms:

   a. nanosecond ____________

   b. millisecond ____________

   c. microsecond ____________

3. Fill in either the appropriate unit of time or the appropriate fraction of a second to complete the following table:

Unit of Time		Fraction of a Second
8 ns	=	__________
__________	=	17/1,000,000
__________	=	364/1000
32 μs	=	__________
500 ms	=	__________
1000 μs	=	__________

SOLUTIONS

1. Write the proper abbreviation for each of the following terms:

   a. microsecond — μs or microsec

   b. nanosecond — ns or nanosec

   c. millisecond — ms or msec

2. Write the equivalent fraction of a second for the following terms:

   a. nanosecond — 1/1,000,000,000

   b. millisecond — 1/1,000

   c. microsecond — 1/1,000,000

3. Fill in either the appropriate unit of time or the appropriate fraction of a second to complete the following table:

Unit of Time		Fraction of a Second
8 ns	=	8/1,000,000,000
17 μs	=	17/1,000,000
364 ms	=	364/1000
32 μs	=	32/1,000,000
500 ms	=	500/1,000
1000 μs	=	1000/1,000,000

EXERCISES

4. Name the function represented by each of the following symbols:

a. / ______________________

b. + ______________________

c. > ______________________

d. < = ______________________

e. * ______________________

f. = ______________________

g. ⩽ ______________________

h. : ______________________

i. >= ______________________

j. * * ______________________

k. < ______________________

l. ⩾ ______________________

m. – ______________________

n. ≠ ______________________

o. ↑ ______________________

SOLUTIONS

4. Name the function represented by each of the following symbols:

a. /	divide
b. +	add
c. >	greater than
d. <=	less than or equal to
e. *	multiply
f. =	equal to or is replaced by
g. ≤	less than or equal to
h. :	compare
i. >=	greater than or equal to
j. * *	raise to the power of
k. <	less than
l. ≥	greater than or equal to
m. –	subtract
n. ≠	not equal to
o. ↑	raise to the power of

EXERCISES

5. Write each of the following expressions using symbols to indicate the specified function:

   a. Y is greater than X ______________________

   b. 2 raised to the fourth power ______________________

   c. 14 times 77 ______________________

   d. A is not equal to B ______________________

   e. R is less than or equal to 7 ______________________

   f. 16 divided by XY ______________________

   g. 12 is less than 17 ______________________

   h. X is compared with A ______________________

SOLUTIONS

5. Write each of the following expressions using symbols to indicate the specified function:

a. Y is greater than X	Y > X
b. 2 raised to the fourth power	2 ↑ 4 or 2**4
c. 14 times 77	14 * 77 (DO NOT USE 14 X 77)
d. A is not equal to B	A ≠ B
e. R is less than or equal to 7	R ≤ 7 or R <= 7
f. 16 divided by XY	16/XY
g. 12 is less than 17	12 < 17
h. X is compared with A	X:A

Take the test for this module and evaluate your answers before studying another module.