



# Beginning RPG 152-115

Student Name _____
Score _____ / 20

## Chapter 3

1 point for each correct answer.

**TRUE/FALSE. Write 'T' if the statement is true and 'F' if the statement is false.**

- \_\_\_\_\_ 1. Binary format for representing numbers, dispenses completely with EBCDIC and stores a number as its direct binary equivalent.
- \_\_\_\_\_ 2. If you list more than one key line, you are specifying a composite or concatenated key.
- \_\_\_\_\_ 3. Logical files describe how data appears to be stored in a database. Logical files do not actually contain the records but store access paths, or pointers, to the records in the physical files.
- \_\_\_\_\_ 4. Field reference files can enforce uniformity and consistency throughout an application system that facilitates program development and maintenance.
- \_\_\_\_\_ 5. Instead of reading long strings of 1s and 0s, most programmers find it easier to represent byte values using hexadecimal (base 16) notation for the high- and low-order bits.
- \_\_\_\_\_ 6. A physical file defined at field level, with one or more of those fields is designated as a key field, can subsequently access records stored in that file in either key sequence or arrival sequence (first-in, first-out). If you do not define a key field, access is limited to arrival sequence.
- \_\_\_\_\_ 7. A logical file based on a single physical file is called a simple logical file.
- \_\_\_\_\_ 8. Logical files actually store data records in arrival sequence—that is, the order in which they are written to the file.
- \_\_\_\_\_ 9. A data type must be assigned to each field in a physical file to determine how the field's values are stored, how much storage the field occupies, and what kinds of operations can be performed on the field when it is used within a program.
- \_\_\_\_\_ 10. If your input file is externally described, you do not need to code Input specifications for the file. When you compile your program, the system will copy the file's definition into your source listing.

**MULTIPLE CHOICE. Choose the one alternative that best completes the statement or answers the question**

- \_\_\_\_\_ 11. Without this file-level keyword, the system permits records with duplicate key values.
- NODUPKEY
  - UNIQUE
  - ABSKEY
  - SINGLE**
- \_\_\_\_\_ 12. You must assign a data type to each field in a physical file to determine how those values are stored. Numeric fields are more complicated. DDS supports which three commonly used data types?
- Binary (type B), zoned decimal (type S) and packed decimal (type A)
  - Zoned decimal (type S), packed decimal (type P) and character decimal (type A)
  - Packed decimal (type P), zoned decimal (type S) and binary (type B)
  - Binary (type B), zoned decimal (type S) and packed decimal (type A)**
- \_\_\_\_\_ 13. Logical files describe how data appears to be stored in the database. They do not actually contain data records but:
- Store access paths, or pointers, to records in physical files
  - Include record format descriptions
  - Contain records with duplicate key values
  - List the key fields in order from major to minor
- \_\_\_\_\_ 14. Data in packed decimal format takes just over \_\_\_\_\_ the amount of storage it would take to store the same number in zoned decimal format because in packed, only the low-order bits of a number are stored.
- one third
  - one fourth
  - one half
  - one fifth**
- \_\_\_\_\_ 15. Level checking is a feature that prevents you from running a program that:
- Creates a physical file to serve as a data dictionary of fields.
  - Does not specify the line positioning on a page of DDS using keywords.
  - Is based on an obsolete or inaccurate definition of a database file.
  - Does not allow the advantage of field reference files**
- \_\_\_\_\_ 16. IBM developed a coding scheme to allow any data character—whether numeric or nonnumeric—to be represented to the computer. This coding scheme assigns a unique eight-bit binary pattern to each representable character and is based on the English alphabet. It is:
- DDS
  - EBCDIC
  - EDTCDE
  - PFILE**

- \_\_\_\_\_ 17. Join logical files:
- a. Give the appearance that the physical files have been merged together.
  - b. Change the definition of a physical file after you have created a logical file based on that definition.
  - c. Join fields from different physical files into a single record using a matching field common to the physical files on which to base the join.
  - d. All of these.
- \_\_\_\_\_ 18. In EBCDIC, lowercase characters sort before uppercase characters, and letters sort before numbers: for example, a, b, c, ... A, B, C, ... 7, 8, 9. In ASCII, the opposite is true...numbers sort before letters, etc. Both are examples of:
- a. Collating sequences.
  - b. High and low order bits.
  - c. Redundancy
  - d. Data dictionaries
- \_\_\_\_\_ 19. Logical files describe how data appears to be stored in the database. Logical files do not actually contain data records. Instead, they store:
- a. Values of key fields.
  - b. Access paths, or pointers, to records in physical files.
  - c. Tables, views, and indexes that were originally created by SQL.
  - d. None of these.
- \_\_\_\_\_ 20. If you define a file only to the record level, any RPG IV program that uses that file must subdivide the record into the appropriate fields in the Input or Output specifications. Because the field descriptions are explicitly coded in the RPG IV programs that will use it, this type of file is called a/an:
- a. Simple logical file
  - b. Join logical file
  - c. Program-described file
  - d. Physical file.