

Programming Languages

MODULE TEST

You may wish to review the exercises or audio-visual material before taking this module test. Once you begin the test, do not refer to the course materials.

There are seven questions.

1. Indicate whether each of the following statements refers to an absolute (A) or a relocatable (R) assembler by writing the correct letter in the space provided.

Statement	Type of Assembler
Converts the source assembly language into an intermediate form called object code, which must be linked to be executed.	_____
Faster than other assemblers and requires fewer resources.	_____
Converts the source assembly language directly into binary machine code.	_____
Several programs can be connected into a single executable program.	_____
All addresses are translated from symbolic to absolute form.	_____
All addresses are translated into a special form and are relative to a single reference address.	_____

2. Indicate whether each of the following statements refers to source (S), object (O), or machine (M) code by writing the correct letter in the space provided.

Statement	Code Type
All addresses are absolute.	_____
Form of a program that a computer can execute.	_____
Must be translated before it can be executed.	_____
Code produced by linkers.	_____
Form in which a programmer writes a program.	_____
Must be linked before it can be executed.	_____
Addresses are in a special relative form.	_____
Expressed in binary format.	_____
Addresses and variable names may be symbolic.	_____
May be saved for later linking with programs assembled or compiled at some other time.	_____
Form of a program produced by relocatable assemblers and compilers.	_____
Form of a program produced by absolute assemblers and interpreters.	_____

3. Indicate whether each of the following statements describes a translator (T), a subroutine (S), or an argument (A) by writing the correct letter in the space provided.

Statement	Software Feature
Sequence of instructions designed to be solved by more than one program.	_____
May require the use of a linker to complete code conversions.	_____
A single program may call this series of instructions more than once.	_____
Address of memory location to be used as source or destination of manipulated data.	_____
Complete series of steps that converts a program from source code to computer-executable code.	_____

4. For each of the following statements, write a T in the space provided if the statement correctly describes macroassemblers. Write an F if it does not correctly describe macroassemblers.

Statement	T or F
Greatly increased operational speed slightly lessens the chance of error-free programs.	_____
Can significantly raise the daily output of assembly language programs.	_____
It has libraries that eliminate duplicate effort from program to program.	_____
During the first pass through the source program, it skips over non-macroinstructions.	_____
Assures a better chance of error-free programs.	_____
Reads and stores each macroinstruction for later expansion according to a specific definition and operands.	_____

5. Indicate whether each of the following characteristics describes a macroassembler (M) or a subroutine (S) by marking the correct letter in the space provided.

Characteristic	M or S
Expanded once for each special assembly language instruction in the program.	_____
Frequently consumes more memory.	_____
Faster assembly because it appears only once in a program.	_____
Slower assembly because each instruction must be expanded.	_____
Slower execution because several operations must be performed each time it is called and returns.	_____
Faster execution because manipulated code is "in-line" with normal code.	_____
Generally more efficient in memory space.	_____
Only one copy of each program, regardless of number of times called.	_____

6. Fifteen statements of characteristics and advantages of two types of translators are given below. Indicate whether each statement refers to compilers (C) or interpreters (I) by writing the correct letter in the space provided.

Characteristic	Type of Translator
Execution of single statement operations. Process continues with next logical statement, not necessarily next sequential one.	_____
Assembly language instructions generated one source statement at a time.	_____
Linking into machine code.	_____
Generation of assembly instructions from operations table.	_____
Statements analyzed and translated to simple operations one at a time.	_____
Optimization of assembly language version.	_____
Analysis of statements and creation of table of simple operations.	_____

Characteristic	Type of Translator
Relocatable assembly of program into object code.	_____
Absolute assembly of assembly instructions for one statement.	_____
On-line debugging error correction.	_____

Advantage**Type of Translator**

Faster execution speed.

—

Greater flexibility.

—

Programmer can interact with program while it is executing.

—

Programs can be saved in either source or object form.

—

Quick development and immediate results.

—

7. Check the appropriate box or boxes next to each statement to indicate which language or languages best answer the statement

FORTRAN COBOL BASIC

- | | | | |
|---|--------------------------|--------------------------|--------------------------|
| a. Originally designed for business applications. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. Originally designed for educational applications. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. Originally designed for scientific/engineering applications. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. Normally compiled. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| e. Normally interpreted. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| f. The easiest to learn. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| g. The most English-like in appearance. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| h. The most efficient for numerical calculations. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| i. Standardized. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| j. The language requiring the least mathematical background. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |