

Programming Languages

EVALUATION SHEET

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.	<u> R </u>
Faster than other assemblers and requires fewer resources.	<u> A </u>
Converts the source assembly language directly into binary machine code.	<u> A </u>
Several programs can be connected into a single executable program.	<u> R </u>
All addresses are translated from symbolic to absolute form.	<u> A </u>
All addresses are translated into a special form and are relative to a single reference address.	<u> R </u>

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

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.	<u>S</u>
May require the use of a linker to complete code conversions.	<u>T</u>
A single program may call this series of instructions more than once.	<u>S</u>
Address of memory location to be used as source or destination of manipulated data.	<u>A</u>
Complete series of steps that converts a program from source code to computer-executable code.	<u>T</u>

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

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

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.	<u>I</u>
Assembly language instructions generated one source statement at a time.	<u>I</u>
Linking into machine code.	<u>C</u>
Generation of assembly instructions from operations table.	<u>C</u>
Statements analyzed and translated to simple operations one at a time.	<u>I</u>
Optimization of assembly language version.	<u>C</u>
Analysis of statements and creation of table of simple operations.	<u>C</u>

Characteristic	Type of Translator
Relocatable assembly of program into object code.	<u>C</u>
Absolute assembly of assembly instructions for one statement.	<u>I</u>
On-line debugging error correction.	<u>I</u>

Advantage	Type of Translator
Faster execution speed.	<u>C</u>
Greater flexibility.	<u>C</u>
Programmer can interact with program while it is executing.	<u>I</u>
Programs can be saved in either source or object form.	<u>C</u>
Quick development and immediate results.	<u>I</u>

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