Brookshear-*Computer Science: An Overview*, 9th edition

Test Bank—Chapter Six (Programming Languages)

Multiple Choice Questions

1. Which of the following is an example of a language that is based on the functional paradigm?

A. LISP B. PROLOG C. C D. C++

2. Which of the following is an example of a language that is based on the object-oriented paradigm?

A. LISP B. PROLOG C. C D. C++

3. Most machine languages are based on the

A. Imperative paradigm B. Declarative paradigm
C. Functional paradigm D. Object-oriented paradigm

4. Which of the following is not a type of statement found in a typical high-level imperative programming language?

A. Imperative statement B. Exclamatory statement
C. Declarative statement D. Comment statement

5. Which of the following does not require a Boolean structure?

A. If-then-else statement B. While loop statement
C. Assignment statement D. For loop statement

6. Which of the following is not a control statement?

A. If-then-else statement B. While loop statement
C. Assignment statement D. For loop statement

7. Which of the following is not a control statement?

A. If-then-else statement B. While loop statement
C. Assignment statement D. For loop statement

8. Which of the following is not a step in the process of translating a program?

A. Executing the program B. Parsing the program
C. Lexical analysis D. Code generation

9. Which of the following is not associated with object-oriented programming?

A. Inheritance B. Resolution C. Encapsulation D. Polymorphism

10. Which of the following is not associated with the concept of data type?

A. Coercion B. Boolean C. Operator precedence D. Strongly typed language

11. Positions within arrays are identified by means of numbers called

A. Indices B. Parameters C. Instance variables D. Constants

12. Which of the following is ignored by a compiler?

A. Control statements
B. Declarations of constants
C. Procedure headers
D. Comment statements

13. Which of the following is not a possible value of the expression $4 + 6 \div 2 - 1$

A. 4 B. 5 C. 6 D. 10

14. Which of the following is not a way of referring to a value in a program?

A. Variable B. Literal C. Constant D. Type

15. Which of the following is the scope of a variable?

A. The number of characters in the variable's name
B. The portion of the program in which the variable can be accessed
C. The type associated with the variable
D. The structure associated with the variable

16. Which of the following is a means of nullifying conflicts among data types?

A. Inheritance B. Parsing C. Coercion D. Code optimization

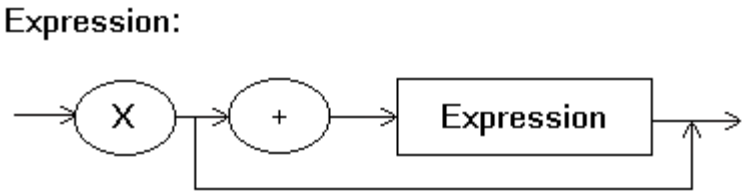
17. Which of the following is not constructed by a typical compiler?

A. Source code B. Symbol table C. Parse tree D. Object program

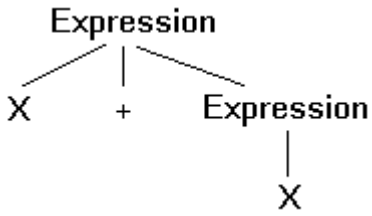
18. Which of the following is a means of defining similar yet different classes in an object-oriented program?

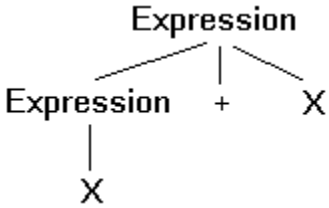
A. Inheritance B. Parsing C. Coercion D. Code optimization

19. Which of the following is not a parse tree of an expression based on the following grammar?



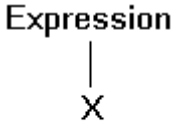
A.





B.

C.



20. Which of the following statements is not a resolvent of the following clauses?

P OR Q OR ¬R ¬P OR T ¬Q OR T R OR T

A. Q OR ¬R OR T B. T OR P C. P OR ¬R OR T D. Q OR T

21. Which of the following can Prolog conclude from the following program?

```
parent(jill, sue).
parent(jill, sally).
parent(john, sue).
parent(john, sally).
sibling(X, Y) :- parent(Z, X), parent(Z, Y).
```

A. `parent(jill, john)` B. `sister(sue, sally)`
C. `sibling(sue, sally)` D. `sibling(jill, sue)`

Fill-in-the-blank/Short-answer Questions

1. In contrast to _____ languages such as English and Spanish, programming languages are considered _____ languages and are rigorously defined by their grammars.

2. List two disadvantages of both machine languages and assembly languages that are overcome by high-level programming languages.

_____

_____

3. Indicate how each of the following types of programming languages is classified in terms of generation (first generation, second generation, or third generation).

A. High-level languages _____

B. Machine languages _____

C. Assembly languages _____

4. List four data types that occur as primitive types in many high-level programming languages.

_____ _____

_____ _____

5. What encoding system is commonly used to encode data of each of the following types?

A. Integer _____

B. Real _____

C. Character _____

6. A _____ array is an array in which all entries are of the same type whereas entries in a _____ array may be of different types.

7. In programming languages that use + to mean concatenation of character strings, the expression

```
"2x" + "3x"
```

will produce what result?

_____

8. Rewrite the following instructions using a single if-then-else statement.

```
    if (X = 5) goto 50
    goto 60
50  print the value of Z
    goto 100
60  print the value of Y
100 . . .
```

9. The following is a program segment and the definition of a procedure named sub.

```
    .
    .
X ← 3;                          procedure sub (Y)
sub (X);                        Y ← 5;
print the value of X;
    .
    .
```

A. What value will be printed by the program segment if parameters are passed by value?

_____

B. What value will be printed by the program segment if parameters are passed by reference?

_____

10. The following is a program segment and the definition of a procedure named sub.

```
    .                           procedure sub
    .                             .
X ← 8;                            .
apply procedure sub;            X ← 2;
print the value of X;             .
    .                             .
    .
```

A. What value will be printed by the program segment if X is a global variable?

_____

B. What value will be printed by the program segment if X is a declared as a local variable within the procedure?

_____

11. To say that a grammar is ambiguous means that _____

_____ .

12. List three items of information that would be contained in a typical parser's symbol table.

_____

_____

_____

13. Give three examples of key words that are often found in high-level imperative or object-oriented languages.

_____ _____ _____

14. In addition to the procedure's name, what other information is contained in a typical procedure header?

_____

15. In the context of the object-oriented paradigm, _____ are templates from which

_____ are constructed. We say that the latter is an instance of the former.

16. In the context of the object-oriented paradigm, a _____ is an imperative program unit that describes how an object should react to a particular stimulus.

17. Based on the sketch of a class definition below, which methods can be invoked from outside an instance of the class?

```
class Example
{public void method1( )
   { . . . }
 private void method2( )
   { . . . }
 public void method3( )
   {…}
 private void method4( )
   { . . .}
}
```

_____

18. What clause would produce the resolvent

P OR R OR S

when resolved with the clause

P OR ¬Q

_____

19. What general rule should be added to the Prolog program below so that Prolog can conclude that ice cream is better than spinach?

```
better(icecream, peanutbutter).
better(peanutbutter, spinach).
```

_____

20. Based on the Prolog program below, what goal should be used to find the siblings of sue?

```
parent(jill, sue).
parent(jill, sally).
parent(john, sue).
parent(john, sally).
sibling(X, Y) :- parent(Z, X), parent(Z, Y).
```

_____

Vocabulary (Matching) Questions

The following is a list of terms from the chapter along with descriptive phrases that can be used to produce questions (depending on the topics covered in your course) in which the students are ask to match phrases and terms. An example would be a question of the form, "In the blank next to each phrase, write the term from the following list that is best described by the phrase."

Term	Descriptive Phrase
assembly language	A step up from machine language
programming paradigm	A program development methodology
structured programming	A methodology that applies well-designed control structures to produce well-organized software
grammar	The rules defining the syntax of a programming language
parse tree	A "pictorial" representation of the grammatical structure of a string
compiler	A program that translates other programs into machine language
interpreter	A program that executes other programs written in a high-level language without first translating them into machine language
high-level language	A notational system for representing algorithms in human compatible terms rather than in the details of machinery
semantics	Meaning as opposed to appearance
syntax	Appearance as opposed to meaning
operator precedence	Dictates the order in which operations are performed
data structure	A conceptual organization of information
parameter	A means of passing information to a procedure or function
data type	Encompasses both an encoding system and a collection of operations
syntax diagrams	A way of representing a grammar
source program	A program expressed in a high-level language

General Format Questions

1. What does it mean to say that a programming language is machine independent?

2. Explain the distinction between the imperative and declarative programming paradigms.

3. Explain why the generation approach to classifying programming languages fails to capture the full scope of today's languages.

4. Explain the distinction between translating a program (in a high-level language) and interpreting the program.

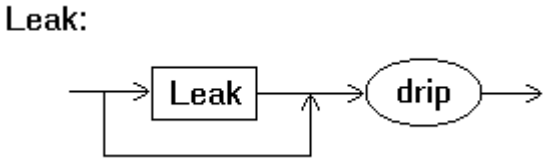
5. Why is the straightforward “goto” statement no longer popular in high-level programming languages?

6. Explain the distinction between a formal parameter and an actual parameter.

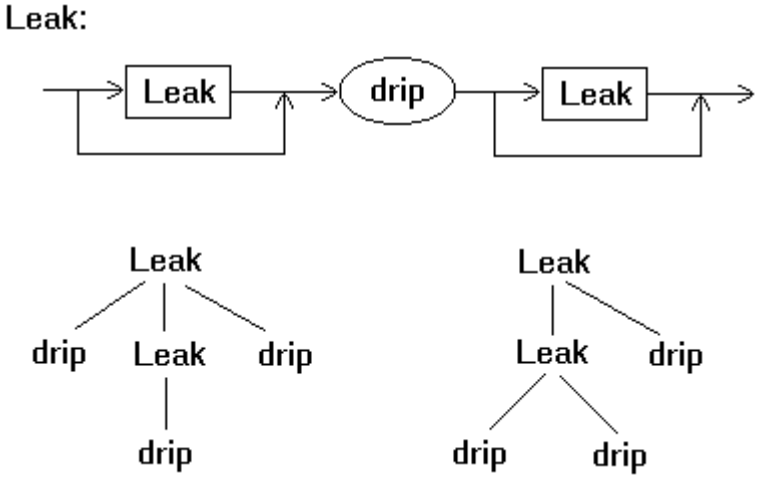
7. Explain the distinction between global and local variables.

8. Explain the distinction between a procedure and a function.

9. Based on the grammar below, draw a parse tree showing that the string “drip drip drip” is a Leak.



10. Show that the grammar below is ambiguous by drawing two distinct parse trees for the string “drip drip drip.”



11. In the context of the object-oriented paradigm, what is a constructor?

12. Briefly describe the task of each of the following.

A. Lexical analyzer

B. Parser

C. Code Generator

13. Explain why key words in a programming language are often reserved words.