

King Fahd University of Petroleum and Minerals

Department of Information and Computer Science

ICS 410-01
(071)

Programming Languages

Final Exam
(120 Minutes)

Dr. Mamdouh Najjar

Name : _____

Student ID : _____

Question No	Maximum points	Student points
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
9	10	
10	10	
11	10	
12	10	
13	10	
Total	130	

January 23, 2008

Question 1:

(10 points)

1.1 What are the four fundamental language evaluation criteria?

**1.2 If you are asked to choose only one evaluation criteria what would it be?
Explain your choice.**

1.3 What is the best programming language up to date? Explain your answer.

1.4 What are the cases for using a compiler and the cases for using an interpreter?

Question 2:

(10 points)

2.1 Write a BNF for a language of the following format:

```
Integer x, y, z
Real i, j, k
x = 3
If ( x >= I ) then
    x = j
endif
print x, j
```

2.2 Why there is a need to describe the meaning of programs?

2.3 What are the main three methods to describe the meanings of programs?

Question 3:

(10 points)

3.1 Describe the role of the lexical analyzer in your implementation of the language Scheme-Lite programming assignment?

3.2 Describe the role of the parser in your implementation of the language Scheme-Lite programming assignment?

Question 4:**(10 points)**

Assume the following program was compiled and executed using static scoping rules:

```
program main;  
  var x : integer;  
  procedure sub1;  
    begin { sub1 }  
    writeln ( 'x = ' , x )  
    end; {sub1 }  
  procedure sub2;  
    var x : integer;  
    begin { sub2 }  
    x := 5;  
    sub1  
    end; { sub2 }  
begin { main }  
x := 10;  
sub2  
end. { main }
```

4.1 What value of x is printed in procedure sub1?

4.2 Under dynamic scoping rules, what value of x is printed in procedure sub1?

Question 5:

(10 points)

5.1 Mark as TRUE or False:

A problem with programming with abstract data types is that the type definitions are all independent and are at the same level.

Inheritance creates obstacles to both the modification problem posted by abstract data type reuse and the program organization problem.

Inheritance provides a framework for the definition of hierarchies of related classes that can reflect the descendant relationships in the problem space.

One advantage of inheritance as a means of increasing the possibility of reuse is that it creates a dependency among the classes in an inheritance hierarchy.

The problem with the reuse of abstract data types is that, in nearly all cases, the features and capabilities of the existing type are not quite right for the new use.

5.2 What are the three characteristic features of object-oriented languages?

Question 6:

(10 points)

6.1: Explain in short the following statement:

"In an FPL, variables are not necessary"

6.2: What is the output of the following *Scheme* programs:

```
(define (y s lis)
  (cond
    ((null? lis) '())
    ((equal? s (car lis)) lis)
    (else (y s (cdr lis))))
  ))
(y 'x '(a b c d e e f f))
(y 'a '(b b a c c s s a a b c))
(y 'y '(a b c x y z))
```

6.3:

```
(define (t3 lista)
  (if (null? lista)
      ()
      (if (null? (cdr lista))
          lista
          (t3 (cdr lista))))
  ))
(t3 '(a b c d))
(t3 '(x y x))
(t3 '(m n n y))
```

Question 7:

(10 points)

7.1 An abstract data type is a data type that satisfies two conditions, complete these two conditions:

- The _____ of, and _____ on, objects of the type are defined in a single syntactic unit
- The representation of objects of the type is _____ from the program units that use these objects, so the only operations possible are those provided in the type's definition

7.2 List three language design issues for abstract data types?

7.3 Given the following statement.

An important benefit of information hiding is increased reliability.

Explain how.

Question 8:

(6 points)

8.1 Complete the following:

Concurrency is naturally divided into instruction level (executing two or more machine instructions simultaneously),

_____ level,

unit level (executing two or more subprograms units simultaneously),

_____ level.

8.2 What are the two levels of concurrency that involve language issues?

8.3 What is the number one motivation for studying concurrency.

Question 9:

(6 points)

9.1 Complete the following:

- An _____ is any unusual event, either erroneous or not, detectable by either hardware or software, that may require special processing.
- The special processing that may be required after detection of an exception is called _____

9.2 List four design issues for exception handling.

Question 10:

(10 points)

10.1 Complete the following:

(a) Symbolic logic can be used for the three basic needs of formal logic:

to express _____

to express _____

and to describe how new propositions can be inferred from other propositions that are assumed to be true.

(b) A _____ can be thought of a logical statement that may or may not be true.

(c) _____ provides a method of expressing collections of propositions.

(d) _____ an inference principle that allows inferred propositions to be computed from given propositions.

(e) _____ finding values for variables in propositions that allows matching process to succeed.

10.2 What are the two essential characteristics of logic programming languages?

Question 11:

(10 points)

11.1 What does the following Prolog program do?

```
test([],0).  
test([X|L], Test) :- test(L, SL), Test is X + SL.
```

11.2 Give an example of a goal to the above program and the result of the goal.

11.3 List three deficiencies of Prolog.

Question 12:

(10 points)

You are asked to design and implement a special purpose language to achieve the following goals:

- handling large amount of data
- do statistical analysis on the data
- generate well formatted reports

In your opinion, what are the design and implementation issues from a language-designer point of view.

Question 13 (bonus):

(10 points)

Course Learning Outcomes:	Upon completion of the course, students will be able to: <ol style="list-style-type: none">1. identify various design issues in programming languages and illustrate with examples how the design issues have been handled in various popular programming languages.2. understand and use formal tools like BNF/EBNF, attribute grammars, operational semantics, etc. to model syntax and semantics of programming languages.3. Understand various design tradeoffs like cost and reliability, efficiency and flexibility, etc. needed to develop a programming languages.4. Show basic program development skill in four programming paradigms: object-oriented, imperative, functional, and logic.
----------------------------------	---

Evaluate your learning outcomes against the above.