King Fahd University of Petroleum and Minerals

Department of Information and Computer Science

ICS 410: Programming Languages

Spring 2006-2007 (062)

Major Exam II: Functional, Logical and Imperative Programming

Date: 5-May-2007 **Time Slot:** 6:30 p.m. – 7:45 p.m.

Duration: 90 minutes **Total Marks:** 150

Name:	**Student ID #:**

Notes:

- Check that you have **six** (6) pages, including this one, containing **four** (4) questions.
- Please skim through all the questions, make sure that you understand them, and then attempt to answer them with a time-allocation in mind. If any question is not clear, get it clarified during the first fifteen minutes.
- If you need to make any reasonable assumptions, please state them clearly as part of your answers.
- There are four questions in this exam each focusing on one of the chapters. You are expected to answer all of them.
- In some questions some parts may have some choices. Clearly identify which selection you decided to do.

Scores:

Question	**Marks**	**Score**
Q.1: Functional Programming Languages	**40**	
Q. 2: Logic Programming Languages	**40**	
Q. 3: Names, Bindings, Type checking and Scope	**40**	
Q. 4: Data Types	**30**	
Total ➔	**150**	

Question 1: (40 marks)

A. [General Concepts of Functional Programming] (15 marks)

Briefly answer **only three** of the following questions:

i. Unlike imperative languages, functional languages are not based on the von Neumann computer. What is the computational model of functional languages and how it is used?

ii. Identify two *key features* and two *key disadvantages* of functional languages.

iii. Briefly explain how does Scheme evaluate a function applied to some parameters?

iv. List three of Scheme predicate functions and explain one of them only.

v. Explain why one may use the Scheme function **`eval`** in his program.

B. [Scheme Program Tracing] (10 marks)

Consider the following Scheme function

```
(define (think x lis)
 (cond
   ((null? lis) 0)
   ((eq? x (car lis)) (+ 1 (think x (cdr lis))))
   (else (think x (cdr lis)))))
```

What does the function **`think`** do? What will be the result of `(think 2 '(3 2 3 2 2))`?

C. [Scheme Program Writing] (15 marks)

Define a Scheme function **`largest`** that receives, as a parameter, a simple list of numbers and returns the largest numbers in the list.

Question 2: **(40 marks)**

A. [General Concepts of Logic Programming] **(15 marks)**

Briefly answer **only three** of the following parts:

i) Logic programming is **non-procedural** programming. Explain what does this mean.

ii) What is a **clausal form** in Logic programming? Give an example.

iii) Explain how we prove theorems using resolution.

iv) Explain why Prolog systems backtrack?

v) What is the closed-world assumption? Why is it considered as a deficiency of Prolog?

B. [Prolog Program Tracing] **(10 marks)**

Consider a database containing the following Prolog statements:

```
student(ali).
good(X):- X>49.
pass(X,Y):- student(X), good(Y).
```

i. What is the result of the query: pass(Ali,70).?

ii. What is the result of the query pass(X, 40)?

C. [Prolog Program Writing] **(15 marks)**

Define a functor ***deleteElement***`(X,S,NS)` with three arguments, an element X and two sets S and NS. NS is constructed from the elements of S by removing all occurrences of X if it is there.

Question 3: *[General Concepts of Variables]* **(40 marks)**

A. Briefly answer **only two** of the following parts: (**10** marks)

i. Indicate the *l*-value(s) and *r*-value(s) in the following statement: x = x + y.

ii. What is the difference between a static and dynamic ancestor of a subprogram SUB?

iii. Explain an advantage and a disadvantage of dynamic type binding.

iv. A variable is to be bound to a value, storage and type. In what sequence must these bindings be made?

B. [Referencing Environment] **(15 marks)**

Consider the following C Program:

```
void fun (void) {
int a, b, c        /* definition 1 */
….
while (…) {
int b, c, d        /*definition 2*/
    ←---------------- 1
while (…) {
int c, d, e        /* definition 3 */
    ←---------------- 2
}
}
    ←---------------- 3
}
```

Show the referencing environment at each of the three marked points in this function.

C. [Dynamic Scoping] **(15 marks)**

Consider the following skeletal C program:

```
void fun1 (void);  /* prototype */
void fun2 (void);  /* prototype */
void fun3 (void);  /* prototype */
void main ( ) {
  int a, b, c;
  …….
  }
void fun1 (void) {
  int b, c, d;
  ……
  }
void fun2 (void) {
  int c, d, e;
  ……
  }
void fun3 (void) {
    int d, e, f;
  …..
  }
```

Assuming that dynamic scoping is used and given the following calling sequence, what variables are visible during execution of the last function called? Include with each visible variable the name of the function in which it was defined: **`main calls fun1; fun1 calls fun2, fun2 calls fun3.`**

Question 4: **(30 marks)**

A. Briefly answer **only three** of the following parts: (**15** marks)

i. What is an enumeration type? Give an example for one. List two advantages for having it.

ii. Explain **two** of the options for string length?

iii. List **three** categories of arrays, and briefly explain one of them stating its advantages and disadvantages.

iv. What are the **two** design issues for records? Briefly explain one of them.

v. What are the **two** problems that may arise when using pointers? Explain one of them.

B. Answer the following two parts about arrays:

i. What is an **access function** for an Array? (**5** marks)

ii. Assume that you are given the declaration **Float Marks [3,5].**

If the elements of the array are stored in the memory using **row-wise** order and the size of a float number is four (**4**) bytes, then what will be the address of the element Marks[3,4] if Marks[1,1] is stored at memory location **1450**. (**10** marks)

Note: For a complete mark you should show how you used the access function.