King Fahd University of Petroleum and Minerals

Department of Information and Computer Science

ICS 313-02
(002)

Fundamentals of Programming Languages

EXAM II
(50 Minutes)

Dr. Mamdouh M. Najjar

Name : ____

ID :______________________________

Question No	Maximum Points	Student Points
1	8	
2	8	
3	8	
4	8	
5	8	
Total	40	

April 23, 2001

<u>Question 1:</u> **(8 points)**

Mark as True or False:

- **Aliasing can occur when pass-by-value-result parameters are used both among two or more parameters and between a parameter and an accessible nonlocal variable.**
- **Local variables in subprograms can be statically allocated, providing support for recursion, or dynamically allocated from a stack, providing efficiency and history-sensitive local variables.**
- **The semantics of an expression is determined in large part by the order of evaluation of operators.**
- **Type conversions can be widening or narrowing. Some narrowing conversions produce erroneous values.**
- **Data-based iterators are loop constructs for processing data structures, such as lists, hashes, and trees.**
- **The conditional branch is the most powerful statement for controlling the flow of execution of a program's statements.**
- **The fundamental idea of an abstract data type is that the use of a type is separated from the representation and set of operations on values of that type.**
- **Implementation methods for data types have no significant impact on their design.**

<u>Question 2:</u> **(8 points)**

Suppose that a language includes user-defined enumeration types and that the enumeration values could be overloaded; that is, the same literal value could appear in two different enumeration types, as in:

```
type
     colors = (red, blue, green, white);
     mood = (happy, angry, blue);
```

Use of the constant `blue` **cannot be type checked.**

Propose a method of allowing such type checking without completely disallowing such overloading. Give an example.

<u>Question 3:</u> **(8 points)**

Let the function FUN **be defined as**

```
function FUN (var K : integer) : integer;
  begin
  K := K + 4;
  FUN := 3 * K - 1
  end;
```

Suppose FUN is used in a program as follows:

```
...
I := 10;
SUM1 := (I / 2) + FUN (I);
J := 10;
SUM2 := FUN (J) + (J / 2);
```

What are the values of SUM1 **and** SUM2

a. **if the operands in the expressions are evaluated left to right?**

Value of SUM1**:**_________________ **Value of** SUM2**:**__________________

b. **if the operands in the expressions are evaluated right to left?**

Value of SUM1**:**_________________ **Value of** SUM2:__________________

<u>Question 4:</u> **(8 points)**

4.1 **Consider the following Pascal case statement. Rewrite it using only two-way selection.**

```
case index - 1 of
    2, 4 : even := even + 1;
    1, 3 : odd := odd + 1;
    0 : zero : = zero + 1;
    else error := true
end
```

4.2 **Rewrite the following code using a loop structure in a language of your choice.**

```
        K := (j + 13) / 27
    loop:
        if k > 10 then goto out
        K := K + 1
        I := 3 * k - 1
        goto loop
    out: . . .
```

<u>Question 5:</u> **(8 points)**

5.1 What are the fundamental semantic models of parameter passing?

5.2 Hand execute the procedure under the following assumptions, and complete the table.

```
proceure BIGSUB;
    integer GLOBAL;
    integer array LIST [1:2];
    procedure SUB (PARAM);
        integer PARAM;
        begin
            PARAM := 3;
            GLOBAL := GLOBAL + 1;
            PARAM := 5;
        end;
    begin
        LIST[1] := 3;
        LIST[2] := 1;
        GLOBAL := 1;
        SUB (LIST[GLOBAL]);
    end;
```

Parameter Passing by	**Contents of LIST[1:2] after the return from SUB**
Value	
Reference	
Name	
Value-result	