2014-2015

Cs 508 MidTerm Papers

[1]

Mr Attitude

Virtual University

2014-2015

[1]

M

CS508 - Modern Programming Languages - Q. No. 1 M - 5
Compare ADA dor loop with dolist and dotime in LISP

Answer: - page 72
Iteration: dotimes and dolist
Apart from recursion, in LISP we can write code involving loops using iterative non recursive mechanism. There are two basic statements for that purpose: dotimes and
dolist. These are discussed in the following paragraphs.
dotimes
dotimesis like a counter-control for loop. Its syntax is given as below:
(dotimes(count n result) body)
It executes the body of the loop n times where count starts with 0, ends with n-1.
The result is optional and is to be used to hold the computing result. If result is given, the function will return the value of result. Otherwise it returns NIL. The value of the count can be used in the loop body.
dolist
The second looping structure is dolist. It is used to iterate over the list elements, one at a time. Its syntax is given below:
(dolist(x L result) body)
It executes the body for each top level element x in L. x is not equal to an element of L in each iteration, but rather x takes an element of L as its value. The value of x can be used in the loop body. As we have seen in the case of dotimes, the resultis optional and is to be used to hold the computing result. If result is given, the function will return the value of result. Otherwise it returns NIL.

Q. No. 2 M - 5
Differentiate between ADA access types and C/C++ pointer type
Answer:- page 52
An access type roughly corresponds to a C++ pointer.
typeAddress_Ref is access Address;
A_Ref := new Address;
A_Ref.Postal_Code := "94960-1234";
Note that, unlike C, there is no notational difference for accessing a record field directly or through an access value.
To refer to the entire record accessed by an access value use the following notation:
Print(A_Ref.all);

Q. No. 3 M - 3
Why predicate is a special function in LISP
Solution:- page 67
Predicates
A predicate is a special function which returns NIL if the predicate is false, T or anything other than NIL, otherwise. Predicates are used to build Boolean expressions in the logical Statements.
The following comparative operators are used as functions for numerical values and return a T or NIL. =, >,<,>=,<=;
For example:

- (= (+ 2 4) (* 2 3))
- T

- (> (- 5 2) (+ 3 1))
- NIL

CS508 - Modern Programming Languages - Q. No. 4 M - 3
Differentiate between C/C++ unions and ADA discriminated type (3)

Solution:- page 52

Discriminated Records

Discriminated records are like union types in C. There are however major differences between C union types and Ada discriminated records. The union type in C is fundamentally unsafe, and therefore unacceptable. For example, consider the following
definition:
union(int, float) puzzle;

Now puzzle has either an intor a float. But at runtime, the compiler and the environment cannot tell which particular value has been stored currently. So we have to trust the programmer. In Ada we are short on trust. The whole philosophy is that a programmer is a human and all humans make mistakes. So to make it safer, Ada uses discriminates in a record that tells what type of data is currently stored there. This is shown in the following
example:

```
typeIF is(Int, Float);
typeInt_Float (V : IF := Int) is record
caseV is
whenInt => Int_Val : Integer;
whenFloat => Float_Val : Float;
end case;
end record;
```

CS508 - Modern Programming Languages - Q. No. 5 M - 2
What additional features are added in COBOL

Answer:- page 27

It was the first language to add macro facility in a high-level language. It also introduced hierarchical data structures(records) and nested selection statements. The language supported long variable names (up to 30 characters) with hyphens, data division, and fixed-point arithmetic.

CS508 - Modern Programming Languages - Q. No. 6 M - 2
Is there anything like templates as in C++.

Answer: - page 60

Generics are like templates in C++ and allow parameterization of subprograms and packages with parameters which can be types and subprograms as well as values and objects.

Question 1 (2 marks)
Whether polymorphism is allowed in Ada or not? Briefly state.
Answer:-

http://archive.adaic.com/docs/present/engle/whyada/tsld021.htm

Ada 83 supported a form of polymorphism through the use of overloading that allowed the developer to reuse a procedural name or operator by redefining the context and semantics. For example, when using integer numbers, the + sign adds one integer to another. The + operator could be redefined to work with an abstract data type consisting of a set of colors and provide the proper color mix (i.e., red + yellow = orange). This polymorphic behavior had to be resolved at compile time. Ada 95 supports a true polymorphic operation through the use of class-wide types and type extensions, which can contextually recognize the appropriate operation to perform.

http://archive.adaic.com/docs/present/engle/whyada/tsld021.htm

Question 2 (2 Marks)

How Bitwise Exclusive Or, and Not is represented in Ada?

Answer:-

Bitwise Exclusive Or =Xor
Bitwise Not =Not
Logical Not =Not

Question 3 (3 Marks)

Identify Local and Global variable

>(defun y-plus(x)(+x y))

Answer:-page 69

This function adds x to y where x is the parameter passed to the function and y is a global variable since it is not defined inside the function. It works in the following manner:
>(setq y 2)
>(y-plus 23)
25
With this we introduce the concept of local and global variables. Local variables are defined in function body. Everything else is global.

Question 4 (3 Marks)

Why Ada does not use '=' operator for assignment like other language?

Answer:- page 53

Assignment statement:-
Like all imperative languages, Ada also supports the assignment statement. However, in Ada the assignment is not an expression like C. The syntax of the assignment statement is as follows:
Variable:= expression;
Note that, in Ada, ':=' is used as the assignment operator whereas '=' is used as assignment operator in C.

Another Solution
Operator Overloading
Ada allows a limited overloading of operators. The exception in Ada is that the assignment operator (:=) cannot be overridden. It can be overridden in case of inheritance from a special kind of "abstract class". When you override the equality operator (=) you also implicitly override the inequality operator (/=).

Question 5 (5 marks)

Briefly explain how LISP evaluates symbolic expression.

Answer:- page 63

symbols: a symbol in LISP is a consecutive sequence of characters (no space). For example a, x, and price-of-beef are symbols. There are two special symbols: T and NIL for logical true and false.
S-expression
An S-expression(S stands for symbolic) is a convention for representing data or an expression in a LISP program in a text form. It is characterized by the extensive use of prefix notation with explicit use of brackets (affectionately known as Cambridge Polish notation).
S-expressions are used for both code and data in Lisp. S-expressions were originally intended only as machine representations of human-readable representation of symbols, but Lisp programmers soon started using S-expressions as the default notation.
S-expressions can either be single objects or atoms such as numbers, or lists.

---

2. ADA case statements in ADA.
Answer:- page54

Case Statements
Case statement is like the switch statement in C, but as compared to C, it is safer and more structured. Its syntax is as shown below:
case expression is
when choice list=>
sequence-of-statements
when choice list=>
sequence-of-statements

---

when others=>
sequence-of-statements
end case;
As opposed to C, the case statement is fully structured; there is no equivalent of the break statement and the control does not fall through tothe next choice after executing the set of statements for the selected choice. The choice _list can have more than one values specified in the form of a range specified with the .. operator like 1..10, discrete values separated by | such as a | e | i | o | u, or a combination of both. The following example elaborates this concept.
caseTODAY is
whenMON .. THU =>
WORK;
whenFRI =>
WORK; PARTY;
whenSAT | SUN =>**;**
REST;
end case;

3. write array in ADA using Days of a week as indexes, just like example given on page: 51

Answer:- page 51

For example if we define Days as below:
type Days **is** (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday);
Then we can create an array which uses Days (or any sub-range of Days) as the array indexes as shown below.
type Daily_Sales is array(Days) ofFloat;

Now the type Daily_Sales is an array type of size 7 and indexes Monday through Sunday. Also note that these literals will be used as indexes in the statements that reference array elements.

4. what is the role of atom in LISP.
Answer:-
Atoms: page 62
Atoms include numbers, symbols, and strings. It supports both real numbers and integers.

5. code written in ADA (procedure), we had to tell the output. 5marks
6. some statements were given, we had to define prolog facts about them 5marks

The proposal of ada is proposed by which company.

Answer:- page 45

The need for a single standard language was felt in 1975 and the draft requirements were given the code name strawman. Strawman was refined to Woodman and then Tinman in 1976. It was further refined to ironman. At that time proposals were invited for the design of a new language. Out of the 17 proposals received, four were selected and given the code names of green, red, blue, and yellow. Initial designs were submitted in 1978 and red and green short listed on the basis of these designs. Standard requirements were then refined to steelman. The designs were refined further and finally Green was selected in 1979. DoD announced that the language will be called Ada. The 1995 revision of Ada (Ada 95) was developed by a small team led by Tucker Taft. In both cases, the design underwent a public comment period where the designers responded to public comments

What is the function paradigm write example function programming language

Answer:- page 63

Functional Programming Paradigm and LISP
Functional programming is a style of programming that emphasizes the evaluation of expressions, rather than execution of commands. The expressions in these languages are formed by using functions to combine basic values. A functional language is a language that supports and encourages programming in a functional style. LISP is a representative language of this style of programming.
Lisp stands for "LISt Process". It was invented by John McCarthy in 1958 at MIT. It has simple data structure (atoms and lists) and makes heavy use of recursion. It is an interpretive language and has many variations. The two most popular versions are Scheme and Common Lisp (de facto industrial standard). It is the most widely used AI programming language.

Write operator which is used in Ada but not used in c/c++

Answer:-see page no 47

Mod, Rem , Abs, ** , .. ,&

Explain briefly the expression of the LISP language.
Answer:-
S-expression
An S-expression(S stands for symbolic) is a convention for representing data or an expression in a LISP program in a text form. It is characterized by the extensive use of prefix notation with explicit use of brackets (affectionately known as Cambridge Polish notation).
S-expressions are used for both code and data in Lisp. S-expressions were originally intended only as machine representations of human-readable representation of symbols, but Lisp programmers soon started using S-expressions as the default notation.

S-expressions can either be single objects or atoms such as numbers, or lists.

1. what is the functionality of KEYWORD INT|CAHR IN snobol4

Answer:- page 36

KEYWORD INT|CAHR IN
This statement assigns the pattern 'INT' | 'CHAR' to the variable KEYWORD.

2.what is error in it >(sqrt x) in LISP
Answer:-page 65

Error: The variable
X is unbound.

3. how many element in it (a (d e)) related to lisp 2
Answer:-
(d e)
See page no 67

4**. what is predicate in a lisp. 3**
Answer:- page 67

Predicates
A predicate is a special function which returns NIL if the predicate is false, T or anything other than NIL, otherwise. Predicates are used to build Boolean expressions in the logical statements. The following comparative operators are used as functions for numerical values and return a T or NIL. =, >,<,>=,<=;
For example:

- (= (+ 2 4) (* 2 3))
- T
- (> (- 5 2) (+ 3 1))
- NIL

For non numeric values you can only check for equality using equal or eq.
Some other useful predicates are listed below:
atom: test if x is an atom
listp: test if x is a list

Which symbol is used for alternation in Snobol?

Answer:- page 36

Alternation:-
Vertical bar is used to specify pattern alternation as shown in the example below.
P1 | P2
This is example of a pattern thatwill match either P1 or P2.
Here are some more examples:
KEYWORD = 'INT' | 'CHAR' This statement assigns the pattern 'INT' | 'CHAR' to the variable KEYWORD.

Overloading in ada and c++?
Answer:- 48
Operator Overloading in ADA
Ada allows a limited overloading of operators. The exception in Ada is that the assignment operator (:=) cannot be overridden. It can be overridden in case of inheritance from a special kind of "abstract class". When you override the equality operator (=) you also implicitly override the inequality operator (/=).

single standard language"

Answer:- page 47

The need for a single standard language was felt in 1975 and the draft requirements were given the code name strawman. Strawman was refined to Woodman and then Tinman in 1976. It was further refined to ironman. At that time proposals were invited for the design of a new language. Out of the 17 proposals received, four were selected and given the code names of green, red, blue, and yellow. Initial designs were submitted in 1978 and red and green short listed on the basis of these designs. Standard requirements were then refined to steelman. The designs were refined further and finally Green was selected in 1979. DoD announced that the language will be called Ada. The 1995 revision of Ada (Ada 95) was developed by a small team led by Tucker Taft. In both cases, the design underwent a public comment period where the designers responded to public comments.

Q.what will be choice of language in functional program paradigm

Anwer:- page 63

Functional Programming Paradigm and LISP

Functional programming is a style of programming that emphasizes the evaluation of expressions, rather than execution of commands. The expressions in these languages are formed by using functions to combine basic values. A functional language is a language that supports and encourages programming in a functional style.
LISP is a representative language of this style of programming.
Lisp stands for "LISt Process". It was invented by John McCarthy in 1958 at MIT. It has simple data structure (atoms and lists) and makes heavy use of recursion. It is an interpretive language and has many variations. The two most popular versions are

Scheme and Common Lisp (de facto industrial standard). It is the most widely used AI programming language.

Q.tell the difference between first,rest,car,cdr

Answer:- see page no 66

List Selectors

In order to select elements from a list, selectors functions are used. There are two basic selector functions known as **first**(or car) and rest(or cdr). The rest can be build with the help of these functions.
first(or car) takes a list as an argument and returns the first element of that list as shown in the following examples:

```
>(first '(a s d f))
a
>(first '((a s) d f))
(a s)
>(setq L '(A B C))
(A B C)
>(car L)
A
```

rest(or cdr) takes a list as its argument and returns a new list after removing the first element from the list. This is demonstrated in the following examples:

```
>(rest '(a s d f))
(s d f).
>(rest '((a s) d f))
(d f)
>(rest '((a s) (d f))
((d f))
>(setq L '(A B C))
(A B C)
>(cdr L)
(B C)
```


Q.tell the out put if

```
defun y-min (x)(- x,y)
 set y 2
 y-min 20
```

Answer:-
-18
See page no 69

MCQs were from book.

Q1 Write two predicate that are used in lisp. Marks 2

Answer:-

Predicates

A predicate is a special function which returns NIL if the predicate is false, T or anything other than NIL, otherwise. Predicates are used to build Boolean expressions in the logical statements. The following comparative operators are used as functions for numerical values and return a T or NIL. =, >,<,>=,<=;

For example:

- (= (+ 2 4) (* 2 3))
- T
- (> (- 5 2) (+ 3 1))
- NIL

For non numeric values you can only check for equality using equal or eq.

Some other useful predicates are listed below:

atom: test if x is an atom

listp: test if x is a list

Also number,symbolp, null can be used to test whether the operand is a number, symbol, or a null value.

Q2 Lists are used in lisp write top elements of the list given as (add 3(multp 3 4)) and (job car school) Marks 3

Answer….See page no 64

The first-level elements in LISP are called top-level elements. For example top elements of list (a b c) are a, b, and c. Similarly, top elements of list (a (b c)) are a and (b c).

An empty list is represented by nil. It is the same as ().

Q3 write a two dimensional array of siz (2x8) in sonobl and intialize it to zero.Marks 3

Answer Page 42:-

The following statement creates a two dimensional array of 2 x 8 and each cell is initialized to NULL string.

N = ARRAY('2,8')

Q4 We want to write a program in ada and want to take marks as inputs from user and subject1=(100)

subject2=(200)

subject3=(400)

How we prevent user to input value that are not in this range

Which is the largest language in which most of the programming done today?

Answer;- 29 doubt
Ada – 1983 – History's largest design effort
It involved a huge design effort, involving hundreds of people, much money, and about eight years. It introduced Packages - support for data abstraction, elaborate exception handling, generic program units, and concurrency through the tasking model.

Explain any unique thing related with lisp conditional control statements. 5

Answer: - page 69
Conditional control: if, when and cond
LISP has multiple conditional control statements. The set includes if, when, and cond. In the following pages we study these statements one by one.
if statement
The if statement has the following syntax:
(if <test> <then> <else>)
That is, an if statement has three parts: the test, the then part, and the else part. It works almost exactly like the if statement in C++. If the test is TRUE then the then part will be executed otherwise the else part will be executed. If there is no else part then if the test is not true then the if statement will simply return NIL. Here is an example that shows the if statement:

```
> (setq SCORE 78)
> 78
> (if (> score 85) 'HIGH
  (if (and (< score 84)(> score 65)) 'MEDIUM 'LOW))
> MEDIUM
```

In the above ifstatement, the thenpart contains 'HIGH and the elsepart is another if statement. So, with the help of nested ifstatements we can develop code with multiple branches.

Why predicate is called a special function in LISP? 3

A predicate is a special function which returns NIL if the predicate is false, T or anything other than NIL, otherwise. Predicates are used to build Boolean expressions in the logical statements.

What do you know about the increment/ decrement or predecessors and successor in enum types in Ada explain with suitable example. 3
Answer:- page 49

Enumeration Types

Just like C, an enumeration type in Ada is a sequence of ordered enumeration literals:
Type Colors is(Red, Orange, Yellow, Green, Blue, Indigo, Violet);
Type State is (Off, Powering_Up, On);
It is however different from C in many respects:
1. There is no arithmetic defined for these types. For example:
S1, S2 : State;
S1 := S1 + S2; -- Illegal
2. One can however add/subtract one (sort of increment and decrement) using the Pred and Succ as shown below:
State'Pred (S1)
State'Succ (S2)

Give at least two examples of math functions in LISP. 2
Answer ;-

page 66
There are many built-in function is LISP. This includes math functions as well as functions for manipulating lists. The math functions include:

+, -, *, /, exp, expt, log, sqrt, sin, cos, tan, max, min

How many types of statements are used for pattern building in SNOBOL. 2
Answer; 37

Pattern
There are two type of statements for pattern building. These are Alternation and Concatenation.
Alternation
Vertical bar is used to specify pattern alternation as shown in the example below.
P1 | P2
This is example of a pattern that will match either P1 or P2.
Here are some more examples:
KEYWORD = 'INT' | 'CHAR' This statement assigns the pattern 'INT' | 'CHAR' to the variable KEYWORD.
Concatenation
Two or more patterns can be concatenated to create new patterns. For example we define P1 and P2 as follows:
P1 = 'A' | 'B'
P2 = 'C' | 'D'
Now if we concatenate these and assign the result to P3 as shown below:
P3 = P1 P2
This will result in assigning 'AC' | 'AD' | 'BC' | 'BD' to P3 which is concatenation of different alternatives of P1 and P2.

MCQS

Anything starting with Capital or underscore is a variable in _____Prolog_____(Prolog, Lisp, Cobol)
In Prolog we specify __Solution____ and not __Problem____ (solution, problem)
LISP is used in _____ and _____(functional paradigm, AI)
First arguments in LISP is _____ (Atom, argument, integer, LIST)
ADA has _No___ do while loop as C/C++ (NO, effective, similar)
Maps in SNOBOL are also available in ____(C, C++, MATLAB, Prolog)
We use _ Unary $___ indirect referencing operator in SNOBOL (Binary ".", Binary $, Unary ".", Unary $)
Elementary types are also called ___ user defines _____ in ADA (Static, user defines, builtin)
ADA is a _____typed language (strongly)
In Snobol 2 spaces are used for , 1st for __pattern matching____, 2nd for ____concatenation_____
__ADA____ has elaborated exception handling (ADA, C++, JAVA, COBOL)
_C__ has a powerful set of operators but poor type checking (C, C++, ADA, LISP)
_____and _____ have declaration paradigm (Prolog, SQL)
1st machine independent language is ___**Algol**____
To make necessary arguments at run time error is called _____(exception handling)
Language with wide domain of application has higher_____(generality)
_____ has distributed computing architecture (COBRA)
Readability has no influence on the cost of _____(deployment)
The depth at which we can think is influenced by ____ **expressive power of the language** _____

Depth at which we can think is influenced by the expressive power of the language.

Best of Luck