Govt . Women Engineering College Ajmer

B.Tech IV Sem I Mid Term 2017-18
Principles of Programming Languages

Time :1 Hour Max Marks : 20

1. Define Attributes of Good Programming Language. (5)
2. Explain Binding And Binding time with Example (5)
3. Explain Elementary Data object and Elementary data type with Example (5)
4. Explin Vector or Array as structured Data type (5)

Ans1.

Attributes of a good language

Clarity, simplicity, and unity - provides both a framework for thinking about algorithms and a means of expressing those algorithms.

Orthogonality -every combination of features is meaningful.

Naturalness for the application - program structure reflects the logical structure of algorithm.

Support for abstraction - program data reflects problem being solved
Ease of program verification - verifying that program correctly performs its required function.

Programming environment - external support for the language.

Portability of programs - transportability} of the resulting programs from the computer on which they are developed to other computer systems.

Cost of use - program execution, program translation, program creation, and program maintenance.

Ans 2.

Binding

Fixing a feature to have a specific value among a set of possible values.

Binding Times

5. **language definition**
6. **language implementation**
7. **translation**
8. **execution**

At language definition - available data types and language structures, e.g in C++ the assignment statement is =, while in Pascal it is **:=**

At language implementation - concerns representation of numbers and arithmetic operations
At translation - Chosen by the **programmer** - variable types and assignments
Chosen by the **compiler** - relative locations of variables and arrays
Chosen by the **loader** - absolute locations

At execution - Memory contents

On entry to a subprogram (copying arguments to parameter locations)
At arbitrary points (when executing assignment statements)

Ans 3.
Elementary data object: contains a data value that is manipulated as a unit.
A data type is a class of data objects with a set of operations for creating and manipulating them.
Examples of elementary data types:
integer, real, character, Boolean, enumeration, pointer.

Specification of a data type

Attributes

Distinguish data objects of a given type Data type and name - invariant during the lifetime of the object. Stored in a descriptor and used during the program execution used only to determine the storage representation, not used explicitly during execution
Values: The data type determines the values that a data object of that type may have Usually an ordered set, i.e. it has a least and a greatest value
Operations
Operations define the possible manipulations of data objects of that type.
Primitive - specified as part of the language definition
Programmer-defined (as subprograms, or class methods)

Implementation of Elementary data type

Storage Representation

Influenced by the hardwareDescribed in terms of: Size of the memory blocks required Layout of attributes and data values within the block .

Ans 4.

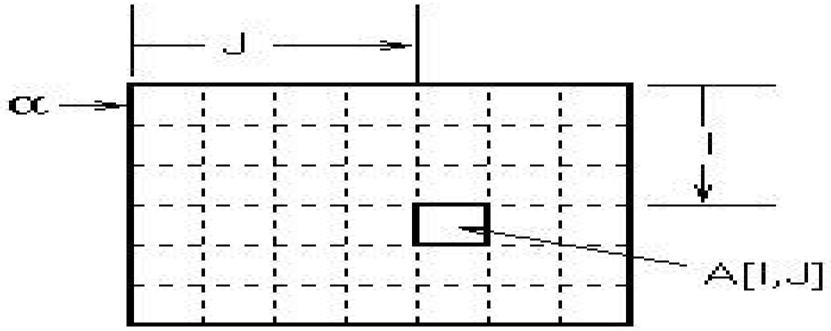
Array

An array is an ordered sequence of identical objects.

The ordering is determined by a scalar data object (usually integer or enumeration data). This value is called the subscript or index, and written as A[I] for array A and subscript I.

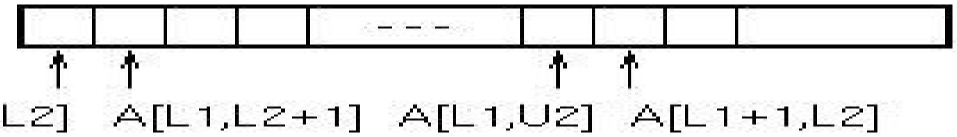
Multidimensional arrays have more than one subscript. A 2-dimensional array can be modeled as the boxes on a rectangular grid.

The L-value for array element A[I,J]is given by the accessing formula on the next slide



L−value(A[I,J]) = α + skip I rows + j columns

Actual storage: A[L1:U1, L2:U2]



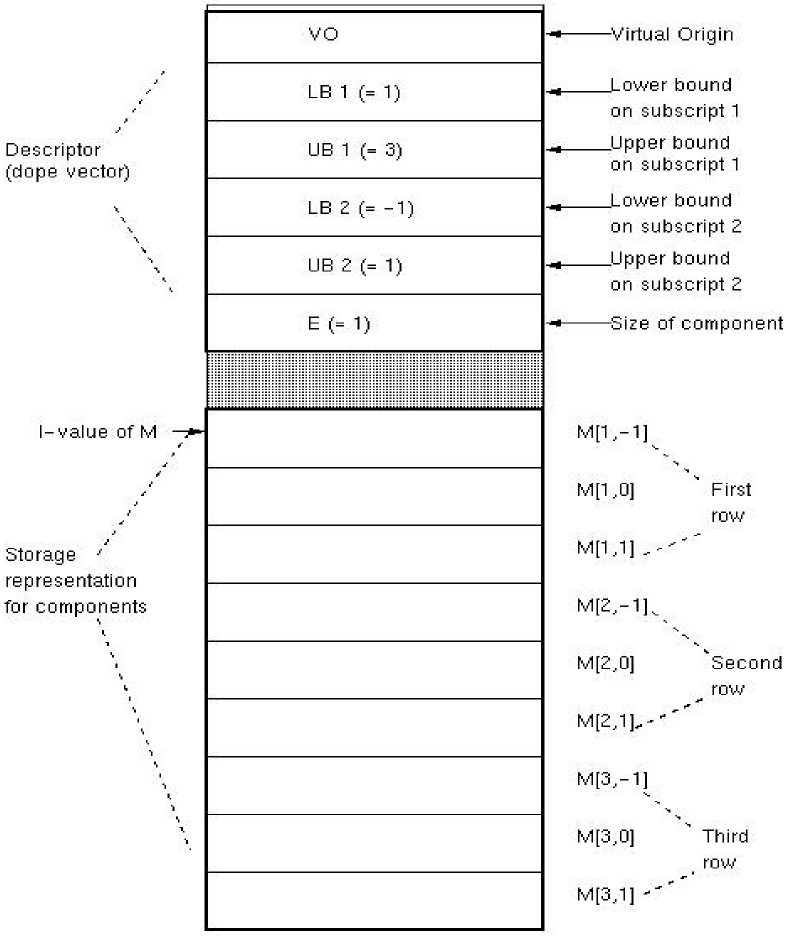
L−value(A[I,J]) = α + number of rows * rowsize
+ number of columns * elementsize

d2 = ele
4
4
1

d1 = Nu — L2 +1) * d2

Rows tc
Column

L−value (J − L2)


VO
LB 1 (= 1)
UB 1 (= 3)
LB 2 (= −1)
UB 2 (= 1)
E (= 1)
Virtual Origin
Lower bound on subscript 1
Upper bound on subscript 1
Lower bound on subscript 2
Upper bound on subscript 2
Size of component
Descriptor (dope vector)
l-value of M
Storage representation for components
M[1,-1]
M[1,0]
M[1,1]
First row
M[2,-1]
M[2,0]
M[2,1]
Second row
M[3,-1]
M[3,0]
M[3,1]
Third row