Introduction

- Records or structure
    - A data structure composed of fixed number of components of different type is termed is record. Component of record are named with identifiers.

Record declaration

- Struct emp
- {
- int id;
- int age;
- float salary;}employee;

- Records are specified by
- 1. number of components.
- Data type of each component.
- Selector used to name each component.

Storage representation of records.

- Number of components like(age,salary,id)
- Data type of each component.
- Selector used to name each component:- to access the value of id selector is employee.id

Storage representation

ID 123
Age 20
Salary 1000

Address calculation

- If the base address of struct emp is 1000.
- Then address of salary can be calculated as
- B+ size of first component+size of second component.

Implementation Variant record

- Implementation of variant record

ID	
age	
salary	
Pay class	
Montly rate	hourly rate
Start date	reg
	overtime

- With help of variant record one record can be represented in multiple forms.

lists

- Components may be homogeneous or heterogeneous

Storage representation in Lisp

Type	Head	tail

- Type field
- Head field
- Tail field
- Type may be atom or list. If the type of field is list, then the first pointer is the head of list. Whereas second list is tail (remaining member)

Sets [mdu may 08]

- Sets are structured data type having unordered arrangement specified by heterogeneous elements and no two elements are of same value.
- Operation on set
- Membership
- Insertion and deletion
- Union,intersection and difference.

Internal representation of sets

- Bit string representation - each and every element of set is represented by bit stringth of length n.
- With this representation insertion into set consist of setting the appropriate bit to 1. and deletion consist of setting the appropriate bit to 0.
- Union and intersection can be done. By bit string reprsentaion.

Hash code representaion of set.

- Scatter stor to place age is done when elements are quite large.
- Hash coding
- X= element to be added into set.
- S= set in which x is to be placed.
- Bx= string reprsenation for x.
- Mx = storage required for set S.
- Hash function
- Ix= address at which we have place x

- For insertion of element , we must ensure
- Whether x is already a memberof set S1. ifnot add to set.
- If x is already in set then it must be stored at that position.