STRUCTURE, UNION & BIT MANIPULATION

Derived Data Types User-defined Data types To extend ou manipulate existing types. . To write new data structures . Defined by the programmer · Derived from existing-types · Memory usage is consistent with base data types Memory usage can vary (eg . unions share memory). eg. struct, union, enum, typedef eg. avezy, pointer, function Structs: Data structures that can store combinations of charactes, integer, floating pt-2 enumerated type data.

A structure is a user-defined data type that groups related variables of different data types under a single name.

SYNTAX: Struct struct_name {

datatype 1 varia;

datatype 2 varia;
... 2.

struct struct name s1;

struct struct-name defines a structure named struct-name. It is a variable of this type with varil, varil etc. fields.

	Dafe Page
	2 ways to declare variables of a struct:
0	Struct Student &
	Char name [50];
	ent age;
	Fort grade; 3 st; /* we declare a variable st*/
2	Struct Student &
	chay name [50]; File to Refer: structs.c
	unt age;
	float grade; 3;
	int main () {
	struct Student SI;
	}
	Accessing Structure members: - 00TPOT:
	printf("Age: %d", sl.age); Age: 40
	S1. age = 20; Age : 20
	S1. age = 20; print ("Age: %d", S1.age); Age: 20
	V
	Typedef
-	Day augation and and a day of the
	For creating synonyms of previously defined data type names.
	eg typedef int length;
	'length' becomes a synonym for the datatype int.
	lig. length a, b, len;
	landth arm [107.

typedy is used in combination with struct to declare as upedef struct Students ? typedef struct Students ? int rollno; chour norme[5], } student; char name [20]; } Student; int main () ? 1/ We can use student like any other type. //Student stud1; Struct Students stud L; I work int main (){ Stud1. roll = 1; stropy (studs name, "John"); Student stud I; } printf ("%od/n", stud 1. rollne); printf ("%sin", stud. name), { typedef struct? to Refer: struct2.c OR8 struct 3. int main () { OUTPUT: Student stud! John Array of Structures Typedef struct Students & int rollno; chay name [5]; } Student; File to Refer: struct 4.0 int main () { // OR struct Students stud [2]; . The above declares an array of like 2. Storage of memory is As any other array, the Contiguous Struct Students ? chai name [so]; { stud [2]; Can refer: union 3-C (w/o using typedes)

fointers to Structures

It is similar to declaring plus to integers, double etc. int *pl;

plu p3 stones address to a struct

he can acces 8 values as: (4p3). Ivoil no

'->' : shorthand operator (*p3). hollno; I same p3-> hollno; I meaning

File to refer: Struct 5.C

- lypedef struct students }
int socl;
Chae name [10]; } Student;

int main () { Student studs; // OR: Struct Students studs;

Studs. roll=1; Stropy (Stud), name, "John");

Student * pl; // ok: Struct Students * pl;

Prints (10/od h)", (* p1).

lup =)

p1 = 8 Steed 1; printf (1% d)n", (*pl) rolling);

" ("%.s\n", (pt). name);
" ("%d\n", pl > roll);

" ("/.s/n", pl-> name); }

output: 1

John

huons Similar to a structure but shares memory for all its members-SYNTAX union rame { datatypes members; data Type 2 member 2; Useful when variables don't need to exist limultaneously, conserving memory. Eg union Data ? int intVal;
float float Val;
their charval; 3; Files to Refer: union 1.C union Data d1; 'union Data' defines a union with an integer, float and char sharing the same memory location. union Student { union Student & int rock; ut soll;

char name [50]; ?; OR char name [50]; ?dl;

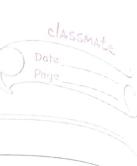
int main () ?

union student student;

Student row = 1;

Stropy (student name, "Mary"); ?

Stropy (student name, "Mary"); ?



Real example on how to use union:

Sike

[Files to Refer: union 3. C]

UNDN STRUCTURE

Memply Each member has unique space All members Share space

Sum of sixes of all members Size of largest number Save memony for single data usage at different times To hold multiple data points Usage