WHAT DO YOU MEAN BY STRUCTURE?

STRUCTURE:

STRUCTURE IS USER DEFINE DATA TYPE. Struct keyword are used to define any structure data type.

ITS ALL MEMBERS ARE OF PUBLIC NATURE BY DEFAULT.

IN C++ WE CAN DECLARE METHODS (FUNCTIONS) inside structure .

ON USING ACCESS MODIFIERS LIKE PRIVATE AND PROTECTED WE CAN CHANGE DEFAULT PUBLIC NATURE OF STRUCTURE.

STRUCTURE CAN BE INHERITE THROUGH ANOTHER STRUCTURE AND ALSO CAN BE INHERIT BY CLASS ES.

IT IS MAINLY USED IN DATA STRUCTURE APPLICATIONS ( LIKE STACK,QUEUE,LINKED LIST, GRAPHS,TREE ETC.)

SIZE OF STRUCTURE VARIABLE IS THE SUM OF ALL DATA ELEMENTS (VARIABLES ) WHICH ARE DECLARE INSIDE STRUCTURE BODY.

SYNTAX OF STRUCTURE:

Struct s1 {

int i; //by default(automatically or implicitly ) public nature BY COMPILER

private:

` int j; //explicit declared as PRIVATE member BY USER.

protected:

int k; //explicit declared as PUBLIC member BY USER

public:

int l; //explicit declared as PUBLIC member BY USER

};

Struct s2 : public s1 { //S2 structure define using s1 structure through inheritance

Int p;

}

Class Abc : public s1 { //Abc class define using s1 structure through inheritance

Public :

Void xyz() {

i=10;

j=20; //Wrong because can’t access private member

k=30;

l=40;

}

};

Void main () {

S2 x;

x.j=110;

x.j=120;

x.k=130;

x.l=140;

x.p=150;

cout<<size of(s1); //size equal to sum of all fields = 8 bytes

}

WHAT DO YOU MEAN BY UNION?

UNION:

Union IS USER DEFINE DATA TYPE. union keyword are used to define any union data type.

ITS ALL MEMBERS ARE OF PUBLIC NATURE BY DEFAULT.

IN C++ WE CAN DECLARE METHODS (FUNCTIONS) inside union .

ON USING ACCESS MODIFIERS LIKE PRIVATE AND PROTECTED WE CAN CHANGE DEFAULT PUBLIC NATURE OF union.

union CAN not BE INHERITE by ANOTHER union AND ALSO CAN not BE INHERIT BY CLASS ES.

SIZE OF union VARIABLE IS THE size OF its biggest DATA ELEMENTS (VARIABLES) WHICH ARE DECLARE INSIDE union BODY. We can use one variable at a time and on using different variable at same time then value of latest will overwrite with previous one.

SYNTAX OF union:

union a1 { //declaration of union with a1 name(Tag)

char c; //by default(automatically or implicitly ) public nature BY COMPILER

private:

` int i; //explicit declared as PRIVATE member BY USER its size is 2 byte

protected:

float f; //explicit declared as PUBLIC member BY USER its size is 4 byte

public:

double d; //explicit declared as PUBLIC member BY USER its size is 8 byte

};

Void main () {

a1 x; //declaration of union variable x

Cout<<sizeof(a1); /\* its size will be 8 byte because union accepts only largest byte which is double in union declaration`\*/

x.c=’A’;

x.i=120;

x.f=13.45;

x.d=1401244987887877897888798899.7898;

}