Home Schooling :

Structure in C

In C programming, a struct (short for "structure") is a user-defined data type that allows you to group different types of data together. It is a way to combine variables of different types into a single unit.

Basic Syntax of a Structure:

```
struct structure_name {
    data_type member1;
    data_type member2;
    // ... more members
};
```

REVIEW

Example of a struct:

```
#include <stdio.h>

// Defining a structure named 'Student'
struct Student {
    char name[50];
    int age;
    float grade;
};

int main() {
    // Creating a structure variable of type 'Student'
    struct Student student1;

    // Assigning values to the structure members
    strcpy(student1.name, "John Doe");
    student1.age = 20;
    student1.grade = 85.5;

    // Accessing and printing structure members
    printf("Name: %s\n", student1.name);
    printf("Age: %d\n", student1.age);
    printf("Grade: %.2f\n", student1.grade);

    return 0;
}
```

Explanation:

- *Defining a Structure: The struct Student defines a structure with three members: name (a string), age (an integer), and grade (a float).*
- *Creating a Structure Variable: The variable student1 is created to store data of type struct Student.*
- *Assigning Values: You can assign values to each member of the structure. Note that for strings, functions like strcpy are used to assign values since you cannot directly assign a string using =.*
- *Accessing Members: To access the members of a structure, use the dot (.) operator, like student1.name or student1.age.*

Why use a struct in C?

- *Organizing data: It helps in organizing and grouping different types of related data together. For example, a "Person" can have a name (string), age (integer), and height (float).*
- *Improved code readability and structure: It makes the program more readable and maintainable.*
- *Handling complex data: It can be used to handle more complex data types like arrays, pointers, and other structures.*

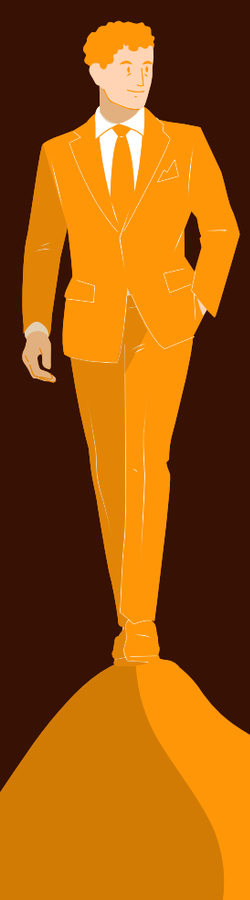
Example of a Simple Structure:

```
#include <stdio.h>

// Defining a structure named 'Book'
struct Book {
char title[100];  // Array to store the title of the book
char author[100]; // Array to store the author's name
int pages;       // Integer to store the number of pages
float price;     // Float to store the price of the book
};

int main() {
// Creating a structure variable 'myBook' of type 'struct Book'
struct Book myBook;

// Assigning values to the members of the structure
strcpy(myBook.title, "C Programming");
strcpy(myBook.author, "Dennis Ritchie");
myBook.pages = 300;
myBook.price = 19.99;

// Accessing and printing the values of the structure members
printf("Book Title: %s\n", myBook.title);
printf("Author: %s\n", myBook.author);
printf("Pages: %d\n", myBook.pages);
printf("Price: %.2f\n", myBook.price);

return 0;
}
```

Explanation:

- *Defining the structure: The structure Book has four members: title, author, pages, and price, each with different data types (char[], int, float).*
- *Creating a structure variable: myBook is a variable of type struct Book that will hold the data for a specific book.*
- *Assigning values: The values are assigned to the structure members, like myBook.title, myBook.author, etc.*
- *Accessing members: The members are accessed using the dot (.) operator.*

“सपनों की उड़ान
वही भर सकते हैं,
जो अपने डर को
पीछे छोड़ देते
हैं।”



REVIEW

PIITR ®
Defining Excellence
PRAGYA INSTITUTE OF IT & RESEARCH
Est.2011