**Structures**

Structures (also called structs) are a way to group several related variables into one place. Each variable in the structure is known as a **member** of the structure.

Unlike an [array](https://www.w3schools.com/c/c_arrays.php), a structure can contain many different data types (int, float, char, etc.).

 create a structure by using the struct keyword and declare each of its members inside curly braces:

example:

struct MyStructure {      
  int myNum;              
  char myLetter;        
};

**Structure Pointer**

A structure pointer is defined as the [pointer](https://www.geeksforgeeks.org/pointers-in-c-and-c-set-1-introduction-arithmetic-and-array/) which points to the address of the memory block that stores a [structure](https://www.geeksforgeeks.org/structures-c/) known as the structure pointer.

**example**

#include <stdio.h>

struct point {

int value;

};

int main()

{

struct point s;

// Initialization of the structure pointer

struct point\* ptr = &s;

return 0;

}

Program to print rollno, name, branch, and batch:

#include <stdio.h>

#include <string.h>

// Creating Structure Student

struct Student {

int roll\_no;

char name[30];

char branch[40];

int batch;

};

// variable of structure with pointer defined

struct Student s, \*ptr;

int main()

{

ptr = &s;

// Taking inputs

printf("Enter the Roll Number of Student\n");

scanf("%d", &ptr->roll\_no);

printf("Enter Name of Student\n");

scanf("%s", &ptr->name);

printf("Enter Branch of Student\n");

scanf("%s", &ptr->branch);

printf("Enter batch of Student\n");

scanf("%d", &ptr->batch);

// Displaying details of the student

printf("\nStudent details are: \n");

printf("Roll No: %d\n", ptr->roll\_no);

printf("Name: %s\n", ptr->name);

printf("Branch: %s\n", ptr->branch);

printf("Batch: %d\n", ptr->batch);

return 0;

}