

Example program for C structure

<http://fresh2refresh.com/c/c-structures/>

This program is used to store and access “id, name and percentage” for one student. We can also store and access these data for many students using array of structures. You can check “[C – Array of Structures](#)” to know how to store and access these data for many students.

```
#include <stdio.h>
#include <string.h>

struct student
{
    int id;
    char name[20];
    float percentage;
};

int main()
{
    struct student record = {0}; //Initializing to null

    record.id=1;
    strcpy(record.name, "Raju");
    record.percentage = 86.5;

    printf(" Id is: %d \n", record.id);
    printf(" Name is: %s \n", record.name);
    printf(" Percentage is: %f \n", record.percentage);
    return 0;
}
```

Output:

```
Id is: 1  
Name is: Raju  
Percentage is: 86.500000
```

Example program – Another way of declaring C structure

In this program, structure variable “record” is declared while declaring structure itself. In above structure example program, structure variable “struct student record” is declared inside main function which is after declaring structure.

```
#include <stdio.h>
#include <string.h>

struct student
{
    int id;
    char name[20];
    float percentage;
} record;

int main()
{
    record.id=1;
    strcpy(record.name, "Raju");
    record.percentage = 86.5;

    printf(" Id is: %d \n", record.id);
    printf(" Name is: %s \n", record.name);
    printf(" Percentage is: %f \n", record.percentage);
    return 0;
}
```

Output:

```
Id is: 1
Name is: Raju
Percentage is: 86.500000
```

C structure declaration in separate header file:

In above structure programs, C structure is declared in main source file. Instead of declaring C structure in main source file, we can have this structure declaration in another file called “header file” and we can include that header file in main source file as shown below.

Header file name – structure.h

Before compiling and executing below C program, create a file named “structure.h” and declare the below structure.

```
struct student
{
int id;
char name[20];
float percentage;
} record;
```

Main file name – structure.c:

In this program, above created header file is included in “structure.c” source file as #include “Structure.h”. So, structure declared in “structure.h” file can be used in “structure.c” source file.

```
#include <stdio.h>
#include <string.h>

struct student
{
    int id;
    char name[20];
    float percentage;
} record;

int main()
{
    record.id=1;
    strcpy(record.name, "Raju");
    record.percentage = 86.5;

    printf(" Id is: %d \n", record.id);
    printf(" Name is: %s \n", record.name);
    printf(" Percentage is: %f \n", record.percentage);
    return 0;
}
```

Output:

Id is: 1 Name is: Raju Percentage is: 86.500000

Uses of C structures:

1. C Structures can be used to store huge data. Structures act as a database.
2. C Structures can be used to send data to the printer.
3. C Structures can interact with keyboard and mouse to store the data.
4. C Structures can be used in drawing and floppy formatting.
5. C Structures can be used to clear output screen contents.
6. C Structures can be used to check computer's memory size etc.

Example

In structures, we have assigned the values to the instances i.e, id, name, percentage in the following way:

```
student1.id=1;  
student2.name = "Angelina";  
student3.percentage = 90.5;
```

Here is the code:

STRUCTUR.C

```
#include <stdio.h>
#include <conio.h>

struct student {
    int id;
    char *name;
    float percentage;
} student1, student2, student3;
int main() {
    struct student st;
    student1.id=1;
    student2.name = "Angelina";
    student3.percentage = 90.5;
    printf(" Id is: %d \n", student1.id);
    printf(" Name is: %s \n", student2.name);
    printf(" Percentage is: %f \n", student3.percentage);
    getch();
    return 0;
}
```

Output will be displayed as:



```
C:\TC\BIN\STRUCTUR.EXE
Id is: 1
Name is: Angelina
Percentage is: 90.500000
```

Printing the student ID and name using struct data type in C program example

Compiler: Visual C++ Express Edition 2005

Compiled on Platform: Windows 2003 Server Standard Edition

Header file: Standard

Additional library: none/default

Additional project setting: Set project to be compiled as C

Project -> your_project_name Properties -> Configuration Properties -> C/C++ ->

Advanced -> Compiled As: Compiled as C Code (/TC)

Other info:

To do: Printing the student ID and name using struct data type

To show: How to use the struct data type in C programming for aggregate data type

```
/* A simple structure program example */
#include <stdio.h>

/* struct definition */
struct student{
char id_num[6];
char name[11];
char gender;
int age;
};

int main(void)
{
/* struct declaration */
struct student studno_1;

/* studno_1.id_num = "A3214"; //Illegal, const char to char[]
// studno_1.name = "Smith"; //Illegal, const char to char[]
*/
printf("First of all, get the size of the struct: %d\n",
sizeof(struct student));
printf("Enter student ID num (5 max): ");
```

```
/* scanf("%s", studno_1.id_num); unsecure version */
scanf_s("%s", studno_1.id_num, 24);
printf("Enter student name (10 max): ");

// scanf("%s", studno_1.name); // old, unsecure version
scanf_s("%s", studno_1.name, 11);
studno_1.gender = 'M';
studno_1.age = 30;
printf("\n-----\n");
printf("ID number: %s\n", studno_1.id_num);
printf("Name : %s\n", studno_1.name);
printf("Gender : %c\n", studno_1.gender);
printf("Age : %d\n", studno_1.age);
printf("-----\n");
return 0;
}
```

Output example:

```
First of all, get the size of the struct: 24
```

```
Enter student ID num (5 max): 2415
```

```
Enter student name (10 max): Maddona
```

```
-----
```

```
ID number: 2415
```

```
Name : Maddona
```

```
Gender : M
```

```
Age : 30
```

```
-----
```

More examples at

<http://www.dailyfreecode.com/Code/example-array-structure-2526.aspx>