1.Programs

|  |
| --- |
| #**include** <stdio.h>  Struct  {  int k;  char c;  };  int main()  {  struct p;  p.k = **10**;  printf("%d\n", p.k);  } |
|  |

Output: Compile time error

Focus on semicolon after structure declared.

Use ‘struct keyword followed by datatype(struct\_name) and variable name.

When initializing variable of structure then value must be separated by semicolon.

Ex:1: struct p x = {**97**,**1**,**3**};

Ex:2. struct p x = {.c = **97**, .k = **1**, .f=**3**};

**Sequence must be followed.**

You can not initialize like this

struct p x = {.c = **97**, .k = **1**, **3**};

If all of the above is not well then compile time error occurred.

2.Program

#**include** <stdio.h>

struct p

{

int k;

char c;

float f;

};

int p = **10**;

int main()

{

struct p x = {**1**, **97**};

printf("%f %d\n", x.f, p);

}

Output: 0.00000 10

Default value for float is 0.0000 and for int is 0000.

In Program 2,we did not initialize float variable ‘f’ so it print 0.0000

Also see next example:

#include <stdio.h>

struct temp

{ int a;} s;

void func(struct temp s)

{

s.a = 10;

printf("%d\t", s.a);

}

main()

{

func(s);

printf("%d\t", s.a);

}-> output: 10 0.(Due to local scope of variable s);

3.Program

#**include** <stdio.h>

struct p

{

int k;

char c;

float f;

};

int main()

{

struct p x = {.c = **97**, .f = **3**, .k = **1**};

printf("%f\n", x.f);

}

Output:3.0000

Also see next

4.Program

#**include** <stdio.h>

struct p

{

int k;

char c;

float f;

};

int main()

{

struct p x = {.c = **97**, .k = **1**, **3**};

printf("%f \n", x.f);

}

Output:0.00000

See next

5.Program

#include <stdio.h>

struct p

{

int k;

char c;

float f;

};

int main()

{

struct p x = {.c = 97, .k = 1, .f=3};

printf("%f \n", x.f);

}

Output: 3.00000

See next

6.Program

#include <stdio.h>

struct p

{

int k;

char c;

float f;

};

int main()

{

struct p x = {.c = 97, .k = 1, .f=3};

printf("%f \n", x.c);

}

Output:0.00000(Default value for char type is 0.0000 and also int)

See next

7.Ptogram

1. #include <stdio.h>
2. struct student
3. {
4. char \*name;
5. };
6. void main()
7. {
8. struct student s, m;
9. s.name = "st";
10. m = s;\\ m.name = s.name;
11. printf("%s%s", s.name, m.name);
12. }

Output: st st

Variable s and m alredy contain name variable.

8.Program:

1. #include <stdio.h>
2. struct student
3. {
4. char \*name;
5. };
6. struct student fun(void)
7. {
8. struct student s;
9. s.name = "alan";
10. return s;
11. }
12. void main()
13. {
14. struct student m = fun();
15. s.name = "turing";
16. printf("%s", m.name);
17. }

Output: Compile time error(identify the scope of variable of ‘S’)

9.Program

1. #include <stdio.h>
2. struct point
3. {
4. int x;
5. int y;
6. };
7. int main()
8. {
9. struct point p = {1};
10. struct point p1 = {1};
11. if(p == p1)
12. printf("equal**\n**");
13. else
14. print

output: Compile time error: we can not directly compare two strcture type variable.

We can compare like this: if(p.x == p1.x)

**Array in Structure:**

#include<stdio.h>

#include<string.h>

struct Cricket

{

char team1[20];

char team2[20];

char ground[20];

};

void main()

{

int i;

struct Cricket match[4];

for(i=0;i<4;i++)

{

printf("Enter Team1");

scanf("%s",match[i].team1);

printf("Enter Team2");

scanf("%s",match[i].team2);

printf("Enter Ground");

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

scanf("%s",match[i].ground);

}

for(i=0;i<4;i++)

{

printf("\n%s Vs %s MATCH=%d CITY=%s ",match[i].team1,match[i].team2,i,match[i].ground);

}

}

Output:

Enter Team1a

Enter Team2b

Enter Groundr

Enter Team1q

Enter Team2w

Enter Groundt

Enter Team1u

Enter Team2

j

Enter Groundh

Enter Team1k

Enter Team2k

Enter Groundj

a Vs b MATCH=0 CITY=r

q Vs w MATCH=1 CITY=t

u Vs j MATCH=2 CITY=h

k Vs k MATCH=3 CITY=j

**Pointer in Structure**

1.C provides a special pointer [**operator**](https://aticleworld.com/operator-in-c-language/), (called arrow) to access a member of a structure pointed to by a pointer variable. The operator is a combination of minus symbol, -, followed by a greater-than symbol, >.

#include<stdio.h>

#include<string.h>

struct Cricket

{

char team1[20];

char team2[20];

char ground[20];

};

void main()

{

int i;

struct Cricket match[2];

for(i=0;i<2;i++)

{

printf("Enter Team1");

scanf("%s",match[i].team1);

printf("Enter Team2");

scanf("%s",match[i].team2);

printf("Enter Ground");

scanf("%s",match[i].ground);

}

struct Cricket \*ptr = match;

for(int i=0;i<2;i++)

{

printf("\n%s Vs %s %s %d",ptr->team1,ptr->team2,ptr->ground,i);

ptr++;

}

}

OUTPUT:

Enter Team1IND

Enter Team2AUS

Enter GroundDELHI

Enter Team1PAK

Enter Team2IND

Enter GroundNAGPUR

IND Vs AUS DELHI 0

PAK Vs IND NAGPUR 1