

# Program

```
// A variable declaration with structure declaration.
struct Point
{
    int x, y;
} p1; // The variable p1 is declared with 'Point'

// A variable declaration like basic data types
struct Point
{
    int x, y;
};

int main()
{
    struct Point p1; // The variable p1 is declared like a normal variable
}
```

# Program

```
struct Point
{
    int x = 0; // COMPILER ERROR: cannot initialize members here
    int y = 0; // COMPILER ERROR: cannot initialize members here
};
```

```
struct Point
{
    int x, y;
};

int main()
{
    // A valid initialization. member x gets value 0 and y
    // gets value 1. The order of declaration is followed.
    struct Point p1 = {0, 1};
}
```

# Program

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

```
struct Point  
{  
    int x, y;  
};
```

```
int main()  
{  
    struct Point p1 = {0, 1};  
  
    // Accesing members of point p1  
    p1.x = 20;  
    printf ("x = %d, y = %d", p1.x, p1.y);  
  
    return 0;  
}
```

# Program

```
#include<stdio.h>

struct Point
{
    int x, y, z;
};

int main()
{
    // Examples of initializtion using designated initialization
    struct Point p1 = {.y = 0, .z = 1, .x = 2};
    struct Point p2 = {.x = 20};

    printf ("x = %d, y = %d, z = %d\n", p1.x, p1.y, p1.z);
    printf ("x = %d", p2.x);
    return 0;
}
```



**Thank you for watching!**

Please leave us your comments.