

STRUCT & UNION

STRUCT DATA TYPE

<p>Simple abstract data type</p>	<pre> struct Color { int r; int g; int b; }; // sizeof(struct Color) == 12 struct Color test; test.h = 255; test.w = 255; test.d=255; // or struct Color test = { 255, 255, 255 }; OR typedef struct { int r; int g; int b; } Color; Color test; test.h = 255; test.w = 255; test.d=255; // or Color test = { 255, 255, 255 }; </pre>
	<pre> typedef struct { int base; int height; int depth; Color c ; } Box; Box Test; Test.base = 50; Test.height = 3; Test.depth = 5; Test.c.r = 255, Test.c.g=255; Test.c.b = 0; or ultimately you can also initialize it at the time when it is declared: Box Test = { 50, 3, 5, {255,255,0} }; </pre>
<p>Structure Array</p>	<pre> typedef struct { int time; float velocity; float mass; } Object; void main() { Object bots[2] = { { 50, 9.5, 10.5 }, { 20, 20, 5.5 } }; for(int i = 0; i < 2; i++) cout << "object-" << i << ": " << ob.velocity/ ob.time)* ob.mass << endl; } </pre>

UNION DATA TYPE

<pre>typedef union { float fVal; int iVal; } MyUnionType;</pre>	<p>sizeof(myUnionType) == 4 , not 8</p> <p>Try the following:</p> <pre>typedef unsigned char ubyte; typedef struct { ubyte r; ubyte g; ubyte b; } Color; typedef union { int code; Color c; } ComboType; ComboType combo; ... in a function, do this : combo.c.r = 255; // or 0xff combo.c.g = 255; // of 0xff combo.c.b = 0; printf("%d", combo.code);</pre>
---	--

Try the following:

- Create a short program to create the union type and initialize the r,g,b only.
- Build and run in debugger. Stop right after you initialize the color.
- Then, also look at the “Watch tab” in the output window.
- Put the parameter and check out the change in the code field. You should experiment by entering different values.

In hex (base-16) display

In Dec (base-10) display

Watch 1	
Name	Value
combo.c.r	0x10 '\x10'
combo.c.g	0x08 '\b'
combo.c.b	0xff 'y'
combo.code	0x00ff0810

or

Watch 1	
Name	Value
combo.c.r	10 '\n'
combo.c.g	8 '\b'
combo.c.b	255 'y'
combo.code	16713738

Watch 1	
Name	Value
combo.c.r	0xff 'y'
combo.c.g	0x0a '\n'
combo.c.b	0x08 '\b'
combo.code	0x00080aff

or

Watch 1	
Name	Value
combo.c.r	255 'y'
combo.c.g	10 '\n'
combo.c.b	8 '\b'
combo.code	527103

ADVANCED TOPICS - DATA ALIGNMENT

In college, this may fall in computer data architecture or compiler course. Different machine architecture does it slightly different. In order to help the CPU fetch data from memory in an efficient manner, data is being arranged in N-bytes chunk, mostly 4-bytes. This is called data alignment.

Every data type has an alignment associated with it which is mandated by the processor architecture rather than the language itself. Aligning data elements improves performance as it allows the processor to fetch data from memory more efficiently.

Examples:

1	<pre>struct TILE { TILE *right; int id; char s[6]; char a; };</pre>	<pre>sizeof (TILE) == 16 , not 15 0 th 4 th 8 th 14 th</pre>
2	<pre>struct TILE { char a; TILE *right; int id; char s[6]; char b; };</pre>	<pre>sizeof (TILE) == 20, not 16 0 th 4 th 8 th 12 th 18 th</pre>
3	<pre>struct TILE { short a; TILE *right; int id; char s[6]; char b; };</pre>	<pre>sizeof (TILE) == 20, not 17 0 th 4 th 8 th 12 th 18 th But note: sizeof(short) == 2</pre>
4	<pre>struct TILE { short a; TILE *right; int id; char s[7]; char b; char c; };</pre>	<pre>sizeof (TILE) == 24, not 19 0 th 4 th 8 th 12 th 19 th 20 th</pre>

EXERCISES :

1) Take social security number : ###-##-####

- 1st 3 numbers : which State
- 2nd 2 numbers : group
- Last 4 numbers : serial number

Write a program using Union structure to allow user to enter a full social security number. Then, you print the region #, group #, and serial number.

Input Display:

Enter your SSN (###-##-####) : **111-22-3456**

Output:

Region: 111

Group : 22

Serial Number : 3456

2) Color Code usually presented in hex, RRGGBB₁₆.

For example: 66CCFF₁₆, i.e. R== 66₁₆, G=CC₁₆, B=FF₁₆, that gives 

Write a function to ask user to input the value of Red, Green, and Blue. Then, it should produce the final color value in Hexadecimals.

Sample function prototype:

```
int createColor( unsigned char red, unsigned char green, unsigned char blue) ;
```

```
// create your own data type ubyte instead of unsigned char
```

Your console output should look like this: (red bold font indicates user input)

```
Enter R: (0 <=x <= 255) : 102
```

```
Enter G: (0 <=x <= 255) : 204
```

```
Enter B: (0 <=x <= 255) : 255
```

```
Output Color : 66ccff
```

After that, go to http://www.w3schools.com/colors/colors_picker.asp to validate the color. You should see this color:



Tips for forming the coder code: use bits shift , such as R<<16 | ...

- 3) Create a function to take in a RGB color code, such as FFFF00, then it should generate the individual R, G, B value. E.g.

Sample function prototype: void makeColor(unsigned int rgb, ubyte *red, ubyte *green, ubyte *blue)

;

Your console output should look like this:

```
Enter the RGB Code : 66ccff

Your output should look like:
R (in hex) = 66 or 102
G (in hex) = cc or 204
  B (in hex) = ff or 255
```

- 4) Modify (3) to populate a Color Structure Type field instead of 3 separate fields:

Sample color structure type:

```
typedef struct {
    int red;
    int green ;
    int blue;
}Color;
```

Your function prototype:

```
void makeColor( int, Color *);
```

```
// 1st parameter is the color code such as 0xff00ff.
```

Go the web page again to validate your conversion.

if you use C++ standard IO : use the “hex” manipulator. E.g. cout << hex << theNumber
if you use C standard IO : use the “0x” manipulator. E.g. printf(“%x”, theNumber);