

### C Program Structure

<code>/* multi-line */</code>	comment
<code>// single-line</code>	comment
preprocessing commands	includes header files, define constants and enum
global declarations	declare global variables
prototype functions	declare functions under main
<code>int main (void)</code>	start of program execution
<code>{</code>	
local declarations; executable statements;	the program
<code>return 0;</code>	end of program
<code>}</code>	
user function()	user defined functions referenced in main
<code>{</code>	
function definition;	
<code>}</code>	

### C Data Keywords

Original K&R Keywords	C90 K&R Keywords	C99 Keywords
int	signed	_Bool
long	void	_Complex
short		_Imaginary
unsigned		
char		
float		
double		

### scanf() Specifiers

<code>%c</code>	character
<code>%d</code>	signed decimal integer
<code>%e, %f, %g, %a<sup>C99</sup></code>	floating-point number
<code>%E, %F, %G, %A<sup>C99</sup></code>	floating-point number
<code>%i</code>	signed decimal integer
<code>%o</code>	signed octal integer
<code>%p</code>	pointer <sup>an</sup> address
<code>%s</code>	string <sup>1</sup>
<code>%u</code>	unsigned decimal integer

### scanf() Specifiers (cont)

<code>%x, %X</code>	signed hexadecimal integer
<sup>1</sup>	begins with first non-whitespace character and ends with the next whitespace character

### printf() Specifiers

<code>%a</code>	floating-point number, hexadecimal digit and P-notation C99/C11
<code>%A</code>	floating-point number, hexadecimal digit and P-notation C99/C11
<code>%c</code>	single character
<code>%d</code>	signed decimal integer
<code>%e</code>	floating-point number, exponent is less than -4 or greater than or equal to the precision
<code>%E</code>	floating-point number, exponent is less than -4 or greater than or equal to the precision
<code>%f</code>	floating-point number, decimal notation
<code>%g</code>	use %f or %e, depending on the value
<code>%G</code>	use %f or %E, depending on value
<code>%i</code>	signed decimal integer <small>same as %d</small>
<code>%o</code>	unsigned octal integer
<code>%p</code>	pointer
<code>%s</code>	character string
<code>%u</code>	unsigned decimal integer
<code>%x</code>	unsigned hex integer, using hex digits 0f
<code>%X</code>	unsigned hex integer, using hex digits 0F
<code>%%</code>	prints a % sign

