ECE 3331 Final Exam Topics

rand( ) and srand( ): how to use them as required in the programming projects.

Chapter 1.

2's comp,
If the msb=0, then the binary value is >=0
If the msb=1, then the binary value is <0
To represent a int >=0, simply set the bits to the desired value
To represent a int <0, simply set the bits to the |desired value|, then flip the bits, and add 1.
A signed7-bit int can hold integer values in the range from –[2^(7-1)] to +[2^(7-1) -1] i.e. -64 to +63
An unsigned7-bit int can hold integer values in the range from 0 to +[2^(7) -1] i.e. 0 to 127

Chapter 2.

while( ) loops
scanf( )
Reading in double values %lf
#include< >
Logical expressions (evaluate as true or false), relational logical operators (==, != &&, ||, <, >, <=, >=)
if(-2.0) is true, if(0.0) is false.
If(x) is true if x != 0.0, else false.
Do-while( ) loops.
Math.h functions (#include<math.h>) **atan( ) vs. atan2(,) and abs( ) vs. fabs( )**
If( ), if-else, if-else-if
How to implement a particular logic using nested if( ) 's, if-else, etc (interpreting someone else's code, or writing your own).
Opening and closing files. File modes "w","r", "w+", "r+" ; **"r+" vs. "w+"**
(checking if the file is successfully opened is in another chapter.)

Chapter 3.

Data types: char, float unsigned short int, etc
the number bytes required to store these variables = the number of memory addresses required to store the variables.
Conversion specifiers for printf( ) and scanf( ): %c, %d, %f, %lf, etc.
In chapt 6, 7, we used %llu is for 8-byte unsigned integers 0 to 2^64-1 (we used this to print out memory addresses converted from pointer values to long long unsigned int values using an explicit cast). Note: the author does not use %llu.
Char variables are stored as integers 0-255 corresponding to the ansi char code.
Special characters '\\', '\"' '\0' string termination (a later chapt)
Arithmetic with mixed data types.
Operators (*,/,+/-) can only operate on operands of the same data type. If two operands are of different data types, the operating system does an implicit cast of the operand of a lower level type to the higher level data type of the other operand.
Left justify and right justify in printf( ), field width, etc.
Macros.
Unary operators: ++, --, pre-increment, pre-decrement, post-increment, post-decrement
Operator precedence and associativity 3+4/5-6*7%8 = ? Use parentheses as needed.
Logical operators (==, != &&, ||, <, >, <=, >=)
for( ) loops
Macro EOF: be clear that the author has a mistake in his use of EOF if scanf( ) vs fscanf( ). We covered this mistake extensively in lecture, and had a programming project that ensured that you understand how and when scanf( ) can return a value EOF = -1 (windows) or EOF=0 (IOS).

Chapter 4.

On the midterm and important:
break, continue, switch( ), conditional expressions
bitwide operators >>, <<, &, |, ~, ^,
sizeof( )

Chapter 5.
Function declaration vs function definition
Using subroutines for actions that get repeated.
How the arguments in the function definition actual declare local variable in the subroutine (not in book).
Local variables in subroutines vs variable in the main( ) function.
Return statements in functions (in main( ) as well as in subroutines)
The preprocessor and preprocessor commands (include, macros, **parameterized macros**)
Preprocessor logic: #ifdef and #endif, #ifndef and #endif, etc..
Note: recursion functions are rarely used in engineering and not on the final exam.

Chapter 6.
Declare array, assign values, retrieve values.
Name of a 1-d array is the memory address of the start of the array.
Declaration of an array and the assignment of values to the cells all in 1 statement.
Strings, character arrays, and '\0'.
char v[100]="string and another string"; stores the while phrase along with the blanks into v[100];
but scanf("%s",v); when the user enters "string and another string" from the keyboard, will only store "string" into v[ ], because it stops reading a string from the keyboard whenever it encounters a blank or tab or newline.
Arrays as arguments to subroutines. "Write a function declaration that takes a float 1-d array as the 1st argument and …."
String handling functions, strstr( ), strcmp( ), strcat( ), strlen( ), etc
2-d arrays. You can declare a 2-d array and load values in the same statement.
2-d arrays as arguments for subroutines.

Chapter 7.1-7.3
Pointer variables.
Write a program segment to use a float pointer to print the value in a float variable to the display.
float x=3.1,*px;
px=&x;
printf("%f",*pf); prints 3.1 to the display
Using pointers as arguments to subroutine.
Pointers and arrays, dereferencing,…

Chapter 7.3-7.8 **(**ECE 3331 only)
Levels of indirection (pointers to pointers, etc)
******************************* from chapter 7
```
int ix=2,y;           // an int value named x
   int *p,*p1;   // two pointers to int variables
   int **  p2pi;     // pointer to pointer to int
   int *** p2p2pi;
   ix = 5;
   p = &ix;  p1=&y; // pointer p now points to int x, and p1 to y
   printf("%p",p);        printf("%p",&x);
   p2pi= &p;
```

Understand the examples in the book regarding memory addresses and pntr-pntr-…
Know how to use both pointer_syntax and array_syntax for accessing cells of a 2-d array.

Chapter 7.4-7.3

Chapter 8. (ECE 3331 only)
Storage classes **static**, auto, and extern
Type qualifiers: **const**, volatile

Chapter 9.
All the possible file modes "w", "w+", "r", "wb", "w+b", etc…
NULL macro.
fseek( ) with SEEK_SET, SEEK_CUR, SEEK_END, feof( ),
gets( ) and fgets( ) read in blanks along with other characters but
scanf("%s",.) and fscanf("%s",.. stop reading at the 1st blank)
Opening files as binary (unformatted) and accessing /using the bytes in the files
as in the programming projects after the midterm.
fwrite( ) fread( )

Chapter 10.
****
Range of possible integer values for bitfield structures

```
struct mystruct{
int i1:3;
unsigned int i2:4;
}
```

e.g. 0<=i2<=15

---

structure variables wherein one of the fields is another structure variable (as in the book).
How structure variable are stored in memory (can be un-used memory addresses between two fields)
Pointer syntax with a pointer to a structure variable.
Enumerated types. The variables are actually 4-byte variables. Know how the
keywords in the declaration of an enumerated variable are assigned int values.
In a question on the final, I might provide a structure variable that is the complete WAVE file header, and ask you to use
the provided structure variable and complete the program to do something with the WAVE file.

Chapter 11.
Run time allocation of 1-d arrays
malloc( ) alloc( )
Freeing allocated memory for 1-d arrays.

ECE 3331 only
Run-time allocation of 2-d arrays
Freeing allocated memory for 2-d arrays.