Know:

1. Software development method
2. General form of a C program
3. Uses of reserved words, standard identifiers, and user-defined identifiers
4. Types of variables, how to declare and assign each one
5. `printf` and `scanf` functions, how to use each part of the functions, use of placeholders (‘%{d,f,c,l,f}’).
6. Use of arithmetic operators (*,/,%,+,–)
7. Know the four types of errors (ch. 2 sec. 8), what they are and what causes them.
9. Conditional and arithmetic operator precedence
10. Forms of the `if` statements (`if`, `if-else`, nested `if`, multiple alternative decisions)
11. `switch` statements
12. Basic math functions (Table 3.1)
13. Function prototypes, definitions, calls, and the difference between them.
14. Writing simple functions
15. `for`, `while`, `do-while` loops
   - Format of loops
   - Uses
   - Steps of defining loops (Initialization, Testing, Update Steps)
   - Be able to write basic loops
Sample problems:

- Problems in book, sample Midterm, quizzes

- Write a function that solves the quadratic equation:

\[ x_1 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]

- What is the difference between these two statements?

(i) \( \text{if}(x > \text{max}) \ \text{max} = x; \)  
\( \text{if}(x < \text{min}) \ \text{min} = x; \)

(ii) \( \text{if}(x > \text{max}) \)  
\( \text{max} = x; \)  
\( \text{else} \)  
\( \text{min} = x; \)

- What is printed in the below for loops?

```c
for(i = 1; i <= 5; printf("\%d\n", i))  
i++;

for(i = 1; i <= 5; i++) {
    printf("\%d\n", i);
    i += 2;
}
```

```c
for(i = 1; i <= 5; i++) ;  
printf("\%d\n", i);  
printf("\%d\n", i);
```

- Find the error(s), if any, in each of these statements:

a. for day=1,3,1  
b. for(day=1,day<3,day++)  
c. for(day=10;day<=20;day++);  
d. for(day=10;day<5;day++);  
e. for(day=100;day<100,day--);  
f. for(day=10;day>100;day++)  
g. for(i=20;i>10;i++) i=i*3;

- Change this while loop into an equivalent for loop:

```c
while(5 > a) { printf("a=%d\n",a); a+=2; }
```