

Some questions may be difficult or tricky. If you can answer these, the exam will be a breeze. Expect Open ended questions like this and some fill in the blank. The second half will be programming questions (i.e., you write code). Some questions will be easier than what is below.

01. What is the scope of a variable?
02. What is the purpose of a function? What makes up a function?
03. Do we want to comment our programs? Why/why not?
04. Is a datatype the same as a function? Why/why not?
05. In C, what is a stream? Why do we want to use streams?
06. How do we control the flow of our program using logic?
07. Can we compare a character type with an integer directly? Why/why not?
08. What type of arguments do we have in a function? What is each type?
09. Should we use floats or doubles more?
10. Is 'a' or 'Z' larger? How do you know?
11. What types of errors do the compiler catch?
12. What types of errors does your IDE catch?
13. When are logic errors caught?
14. When will you know if you dereference NULL?
15. Is it better to use fscanf or scanf?
16. What is the purpose of main()?
  - a. What is/are the return type(s)? List all possible outcomes.
  - b. What is/are the parameter(s)? List all possible outcomes.
  - c. Do we need main() in every C program?

17. How does `FILE *infile` differ from `FILE infile`? Is one better over the other?
18. What is the star operator? When do we use it?\*
19. What are define statements?
20. What are include statements?
21. What all do we need to include for a function prototype?

\* Indicates complete comprehension is not required for Exam 1. We only need to understand what it means for `FILE*`.

## What is wrong with this code?

### MAIN.C:

```
#include "stdio.h"

#include <myHeader.h>

int main(void) {

    getCharFromUser();
    valid = validateChar();

    if (valid == 1)
        printf("SUCCESS! Good Entry!\n");
        printf("valid = %lf", valid);
    else (valid == 0)
        printf("FAILURE! Bad Entry!\nvalid = %lf", valid);

}
```

### Snippet from MYHEADER.H: (Assume this file is proper)

```
char getCharFromUser(void);

int validateChar(char);
```

## Scoping

Call: `myMethod(10, 20, 30);`

```
static int d = 40;

void myMethod(int a, int b, int c) {
    // POSITION 1
    d = 4444;
    if (a + b == 12) {
        int a = 1;
        // POSITION 2
    } else if (c - b == 23) {
        int c = 3;
        // POSITION 3
        {
            c = 33;
            b = 2;
            // POSITION 4
        }
    } else {
        int d = 4;
        {
            d = 44;
            {
                extern int d;
                d = 444; // This references the
global variable per `extern int d;`
                // POSITION 5
            }
            // POSITION 6
        }
        // POSITION 7
    }
    // POSITION 8
}
```

(Questions on next page)

**What is the value of a, b, c, and d at the given positions?**

POSITION 1) a = \_\_\_\_\_ b = \_\_\_\_\_ c = \_\_\_\_\_ d = \_\_\_\_\_

POSITION 2) a = \_\_\_\_\_ b = \_\_\_\_\_ c = \_\_\_\_\_ d = \_\_\_\_\_

POSITION 3) a = \_\_\_\_\_ b = \_\_\_\_\_ c = \_\_\_\_\_ d = \_\_\_\_\_

POSITION 4) a = \_\_\_\_\_ b = \_\_\_\_\_ c = \_\_\_\_\_ d = \_\_\_\_\_

POSITION 5) a = \_\_\_\_\_ b = \_\_\_\_\_ c = \_\_\_\_\_ d = \_\_\_\_\_

POSITION 6) a = \_\_\_\_\_ b = \_\_\_\_\_ c = \_\_\_\_\_ d = \_\_\_\_\_

POSITION 7) a = \_\_\_\_\_ b = \_\_\_\_\_ c = \_\_\_\_\_ d = \_\_\_\_\_

POSITION 8) a = \_\_\_\_\_ b = \_\_\_\_\_ c = \_\_\_\_\_ d = \_\_\_\_\_

**Write a program that:**

- 1) Opens a file called "input.csv" for reading.
  - 2) Opens a file called "output.log" for writing.
  - 3) Read the first 20 characters [assume all characters are lowercase] \$CHAR is the character read.
    - a. If it is a vowel (a, e, i, o, u), then print "\$CHAR is a vowel" to output.log.
    - b. Otherwise, print "\$CHAR is NOT a vowel" to output.log.
    - c. Do not write 20 different statements to read and output.
  - 4) Clean up variables as needed.
  - 5) Exit with code 0 if clean, exit with code 1 if there was a problem.
- Declare any functions directly above main(). [AKA no protocols are needed]
  - No comments needed, only code
  - Assume stdio.h is included. Do not use any other header files.

**Write a function that checks if a number is an Armstrong Number.**

**That is,  $153 = 1^3 + 5^3 + 3^3$ . Each digit of the number cubed is the given number.**

- Use scanf to get an input, In this example 153.
- Then, go digit by digit and sum each cube to determine if it is an Armstrong Number.

```
int main(void) {  
    int input;  
  
    // Define variables as needed  
  
    scanf("%d", &input);  
  
    // Implement  
  
    // Print if it is an Armstrong Number or not  
  
}
```

**Write a function that checks if a number is a Perfect Number.**  
**e.g., N = 28; N's divisors are 1, 2, 4, 7, 14.  $28 = 1 + 2 + 4 + 7 + 14$ . All divisors of the input sums to that input.**

- Use scanf to get an input, in this example 28.
- Then, check every number from 1 to N if it is a divisor.
  - o A divisor can be determined by (  $N \% i == 0$  ) where i varies from 1 to the N.

```
int main(void) {
    int input;

    // Define variables as needed

    scanf("%d", &input);

    // Implement

    // Print if it is a Perfect Number or not

}
```