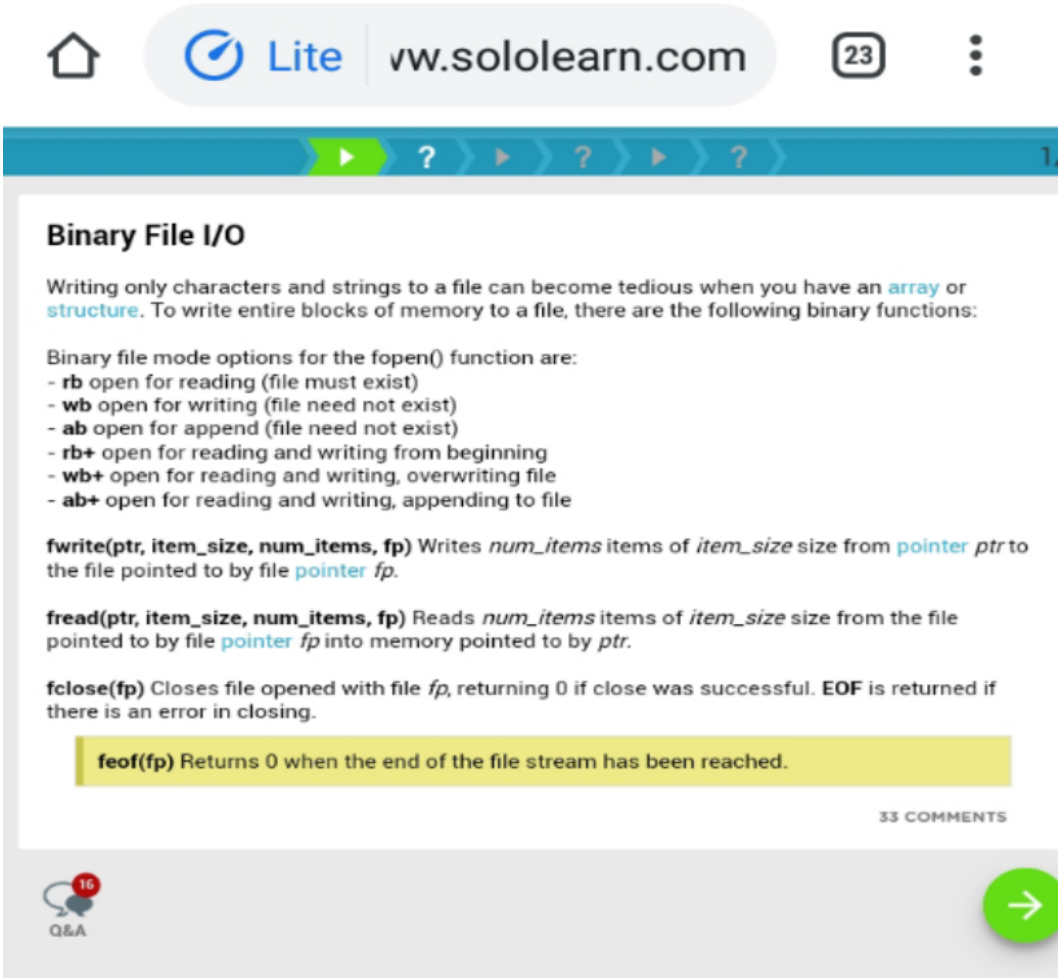
DAILY ASSESSMENT FORMAT

Date:	**20/06/2020**	**Name:**	Nichenametla Bhargavi
Course:	C Programming	**USN:**	**4AL17EC061**
Topic:	**1: Files & Error Handling** **2: The Processors**	**Semester & Section:**	**6th Sem A Sec**
Github Repository:	Bhargavi_Nichenametla		

SESSION

Image of the session



Report – Report can be typed or hand written for up to two pages.

Exception Handling:

- Central to good programming practices is using error handling techniques.
- Even the most solid coding skills may not keep a program from crashing should you forget to include exception handling.
- An exception is any situation that causes your program to stop normal execution.
- Exception handling, also called error handling, is an approach to processing runtime errors.
- C does not explicitly support exception handling, but there are ways to manage errors:-

1. Write code to prevent the errors in the first place. You can't control user input, but you can check to be sure that the user entered valid input.
2. When performing division, take the extra step to ensure that division by 0 won't occur.

Use the exit statement to gracefully end program execution. You may not be able to control if a file is available for reading, but you don't need to allow the problem to crash your program.
Use errno, perror(), and strerror() to identify errors through error codes.

The "perror and strerror" Functions:

When a library function sets errno, a cryptic error number is assigned. For a more descriptive message about the error, you can use perror(). You can also obtain the message using strerror() in the string.h header file, which returns a pointer to the message text.

perror() must include a string that will precede the actual error message. Typically, there is no need for both perror() and strerror() for the same error, but both are used in the program below for demonstration purposes:

```
FILE *fptr;
errno = 0;

fptr = fopen("c:\\nonexistantfile.txt", "r");
if (fptr == NULL) {
  perror("Error");
  fprintf(stderr, "%s\n", strerror(errno));
  exit(EXIT_FAILURE);
}
```

File Handling in C:

- So far the operations using C program are done on a prompt / terminal which is not stored anywhere. But in the software industry, most of the programs are written to store the information fetched from the program.
- One such way is to store the fetched information in a file. Different operations that can be performed on a file are:

1. Creation of a new file (fopen with attributes as “a” or “a+” or “w” or “w++”).
2. Opening an existing file (fopen).
3. Reading from file (fscanf or fgets).
4. Writing to a file (fprintf or fputs).
5. Moving to a specific location in a file (fseek, rewind).
6. Closing a file (fclose).

Certificate for completion of C Programming: