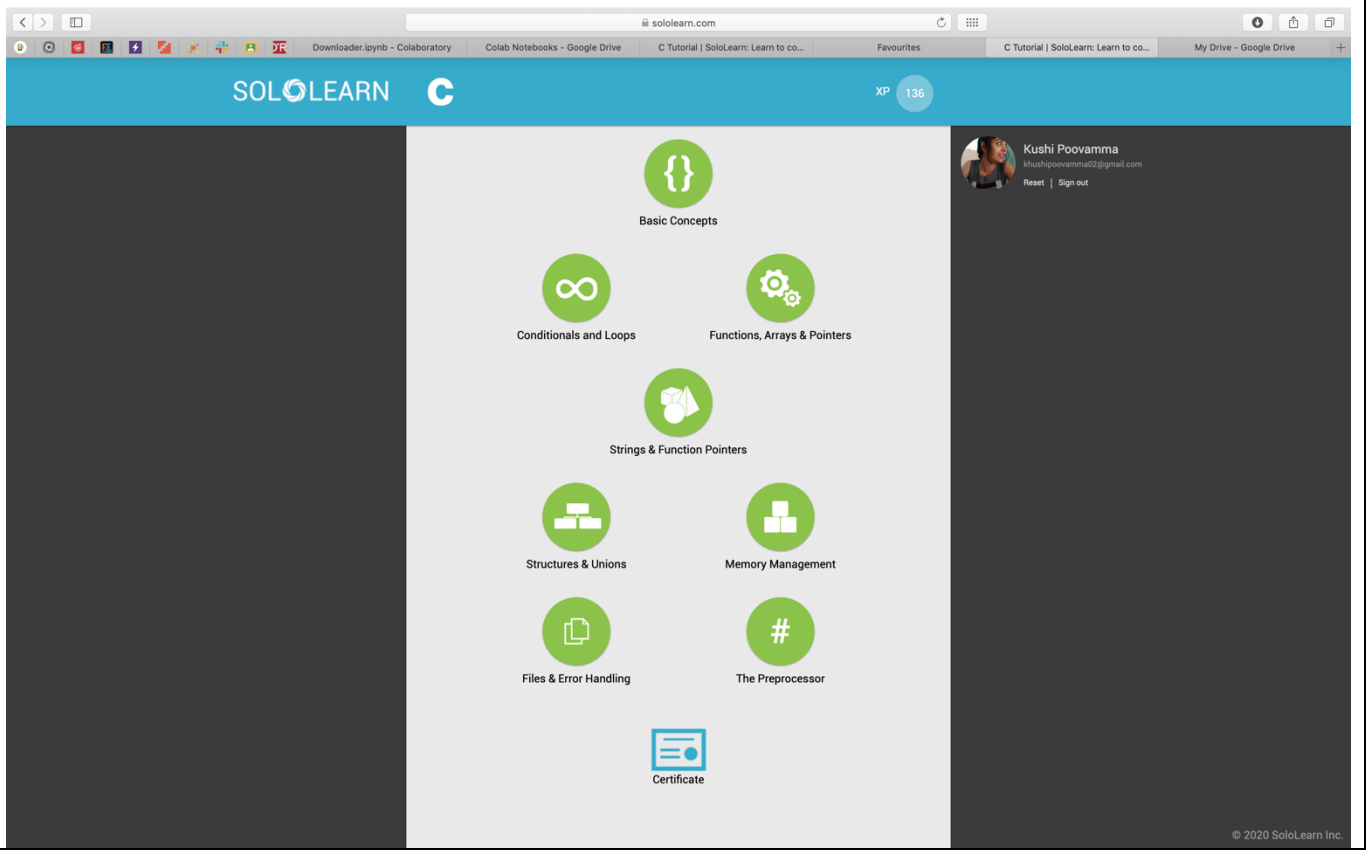
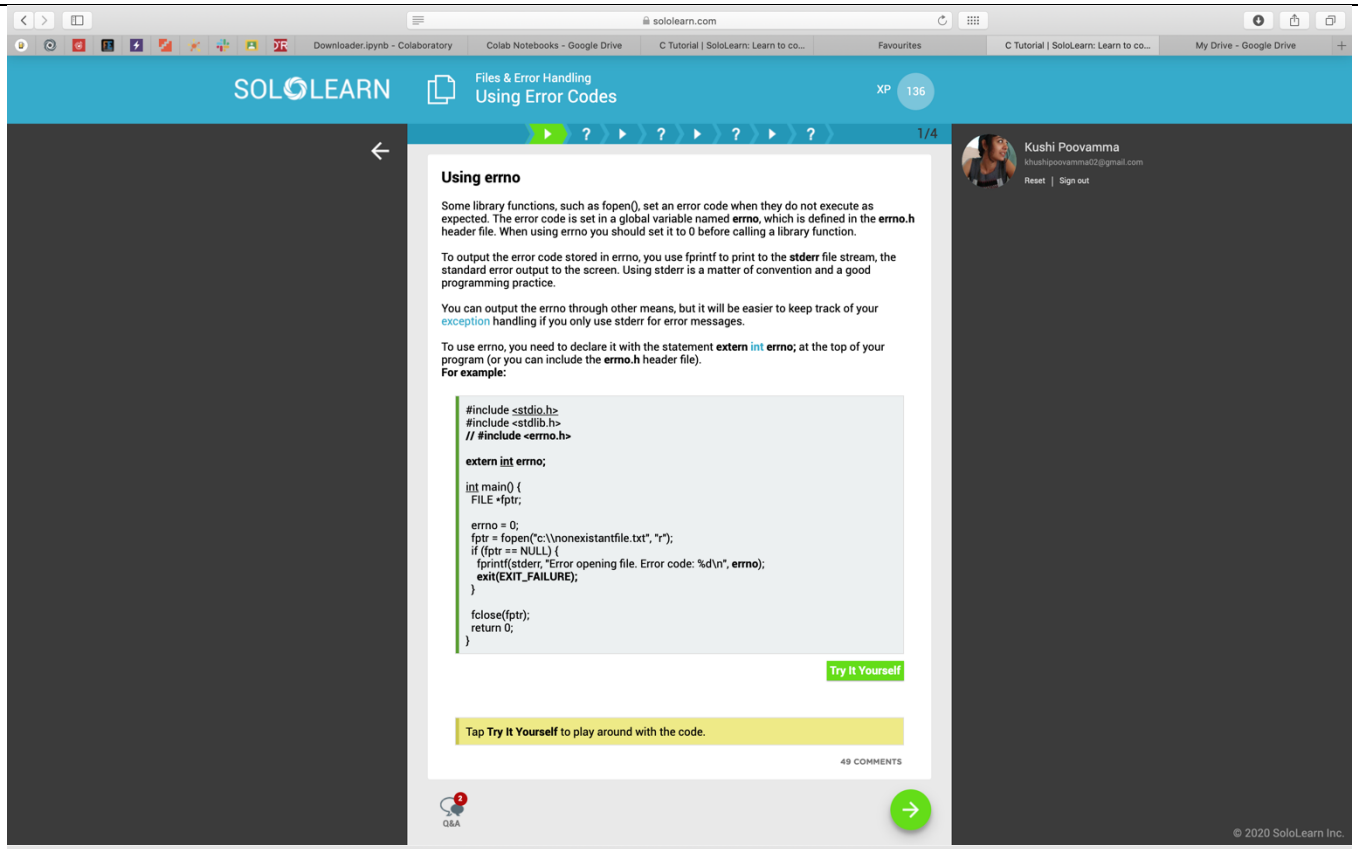
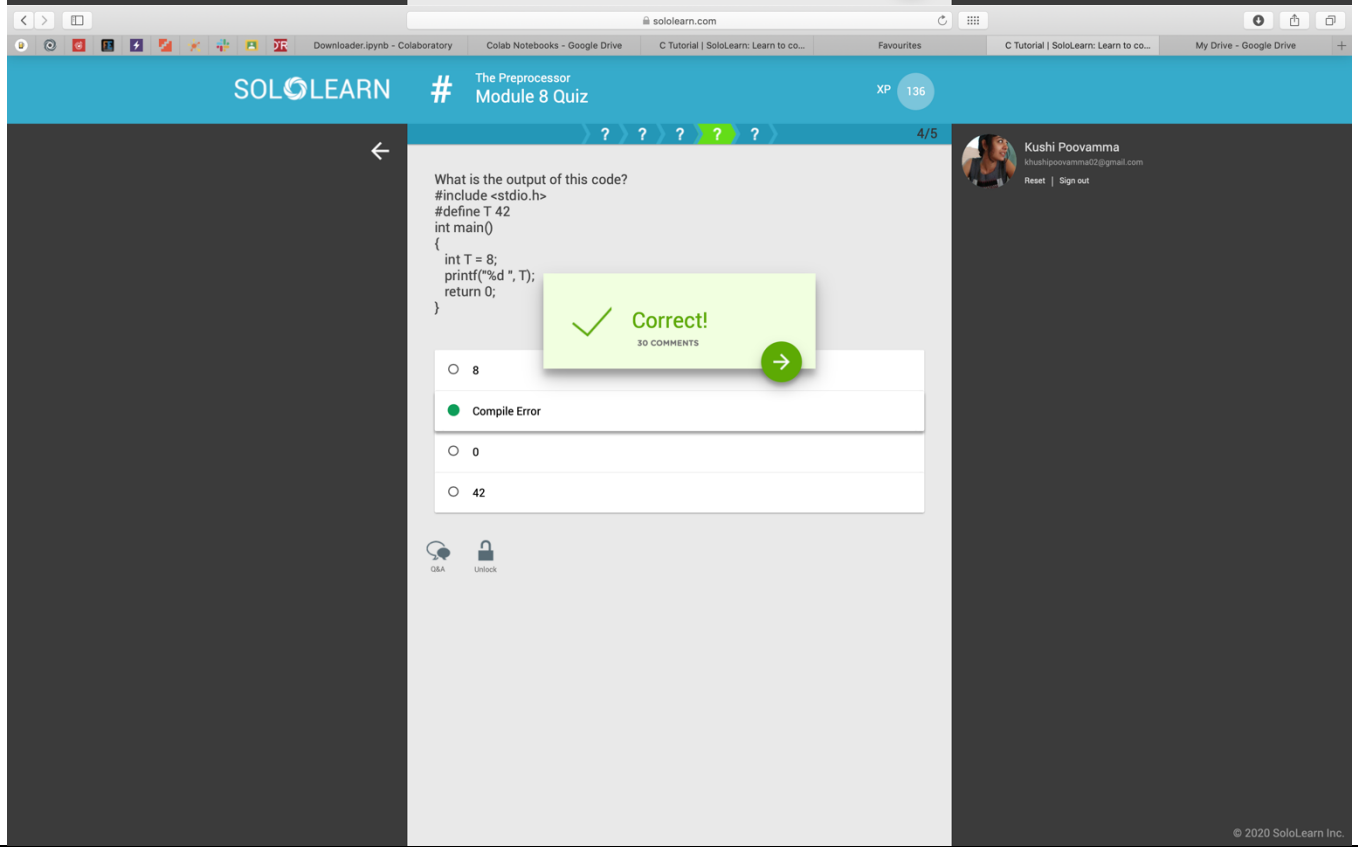
DAILY ASSESSMENT FORMAT

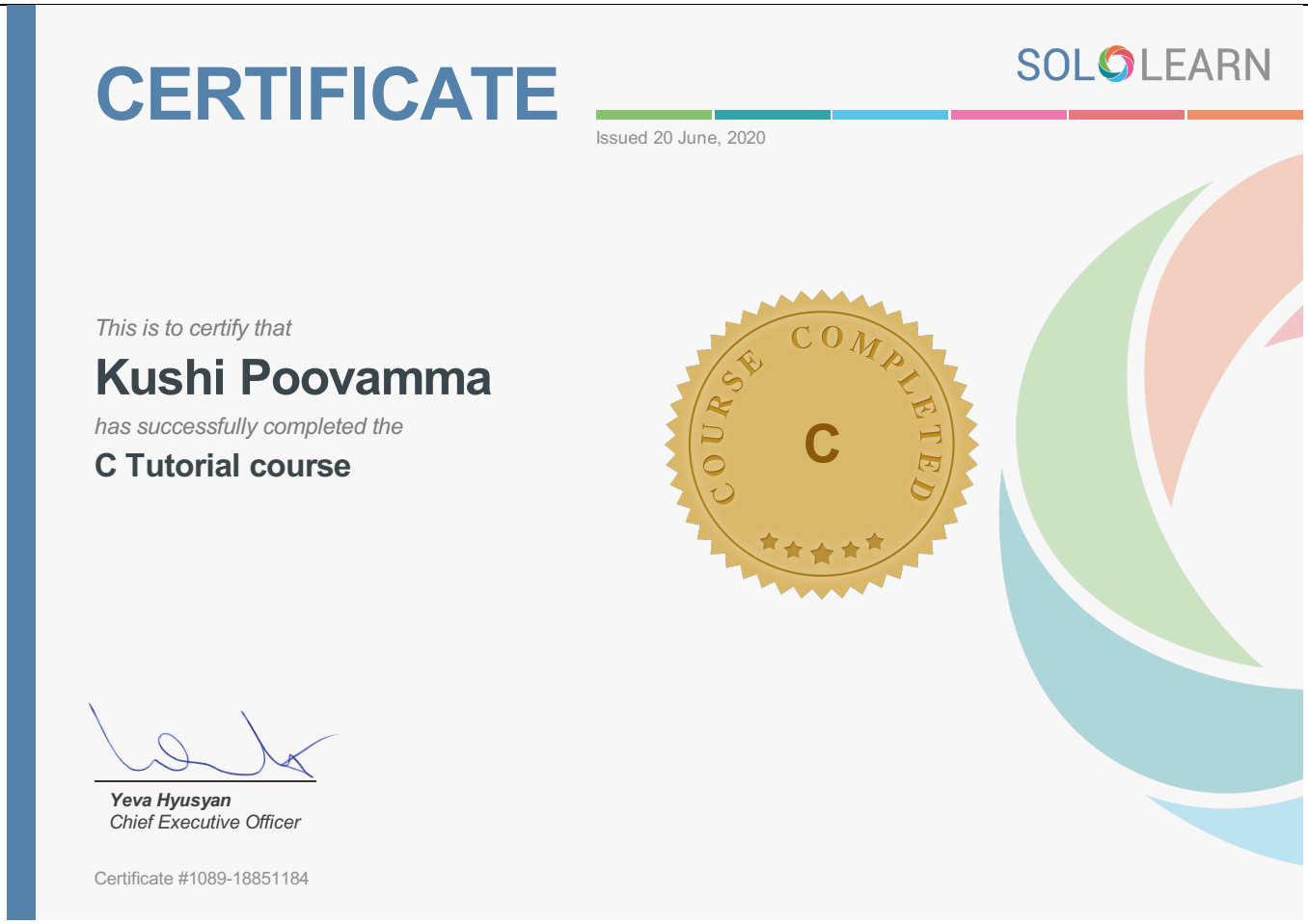
Date:	**20/06/2020**	**Name:**	**K B KUSHI**
Course:	**C Programming ng**	**USN:**	**4AL17EC107**
Topic:	**Files & Error Handling** **The Preprocessor**	**Semester & Section:**	**6 B**
Github Repository:	**https://github.com/alvas-education-foundation/KUSHI-COURSES.git**		

FORENOON SESSION DETAILS				
Github Repository:	**https://github.com/alvas-education-foundation/KUSHI-COURSES.git**			
FORENOON SESSION DETAILS				
Image of session				




sololearn.com
Downloader.ipynb - Colaboratory
Colab Notebooks - Google Drive
C Tutorial | SoloLearn: Learn to co...
Favourites
C Tutorial | SoloLearn: Learn to co...
My Drive - Google Drive
SOLOLEARN
Files & Error Handling
Using Error Codes
XP 136
1/4
Kushi Poovamma
khushipoovamma02@gmail.com
Reset | Sign out
Using errno
Some library functions, such as fopen(), set an error code when they do not execute as expected. The error code is set in a global variable named errno, which is defined in the errno.h header file. When using errno you should set it to 0 before calling a library function.
To output the error code stored in errno, you use fprintf to print to the stderr file stream, the standard error output to the screen. Using stderr is a matter of convention and a good programming practice.
You can output the errno through other means, but it will be easier to keep track of your exception handling if you only use stderr for error messages.
To use errno, you need to declare it with the statement extern int errno; at the top of your program (or you can include the errno.h header file).
For example:
#include <stdio.h>
#include <stdlib.h>
// #include <errno.h>
extern int errno;
int main() {
FILE *fptr;
errno = 0;
fptr = fopen("c:\\nonexistantfile.txt", "r");
if (fptr == NULL) {
fprintf(stderr, "Error opening file. Error code: %d\n", errno);
exit(EXIT_FAILURE);
}
fclose(fptr);
return 0;
}
Try It Yourself
Tap Try It Yourself to play around with the code.
49 COMMENTS
Q&A
© 2020 SoloLearn Inc.


sololearn.com
Downloader.ipynb - Colaboratory
Colab Notebooks - Google Drive
C Tutorial | SoloLearn: Learn to co...
Favourites
C Tutorial | SoloLearn: Learn to co...
My Drive - Google Drive
SOLOLEARN
The Preprocessor
Module 8 Quiz
XP 136
4/5
Kushi Poovamma
khushipoovamma02@gmail.com
Reset | Sign out
What is the output of this code?
#include <stdio.h>
#define T 42
int main()
{
int T = 8;
printf("%d ", T);
return 0;
}
Correct!
30 COMMENTS
8
Compile Error
0
42
Q&A
Unlock
© 2020 SoloLearn Inc.



Report –

Error Handling

Error handling features are not supported by C programming, which is known as exception handling in C++ or in other OOP (Object Oriented Programming) languages. However, there are few methods and variables available in C's header file *error.h* that is used to locate errors using return values of the function call. In C, the function return *NULL* or *-1* value in case of any error, and there is a global variable *errno* which sets the error code/number. Hence, the return value can be used to check error while programming.

```
/* Divided By zero Error i.e. Exception*/#include <stdio.h>

#include <stdlib.h>


void main() {

  int ddend = 60;
```

```
  int dsor = 0;
  int q;

  if( dsor == 0){
    fprintf(stderr, "Division by zero! Exiting...\n");
    getch();
    exit(-1);
  }
  q = ddend / dsor;
  fprintf(stderr, "Value of quotient : %d\n", q);
  getch();
  exit(0);
}
```

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)**

How a Preprocessor works in C?

A Preprocessor is a system software (a computer program that is designed to run on computer's hardware and application programs). It performs preprocessing of the High Level Language(HLL). Preprocessing is the first step of the language processing system. Language processing system translates the high level language to machine level language or absolute machine code(i.e. to the form that can be understood by machine).

- **The preprocessor doesn't know about the scope rules of C. Preprocessor directives like #define come into effect as soon as they are seen and remain in effect until the end of the file that contains them; the program's block structure is irrelevant.**