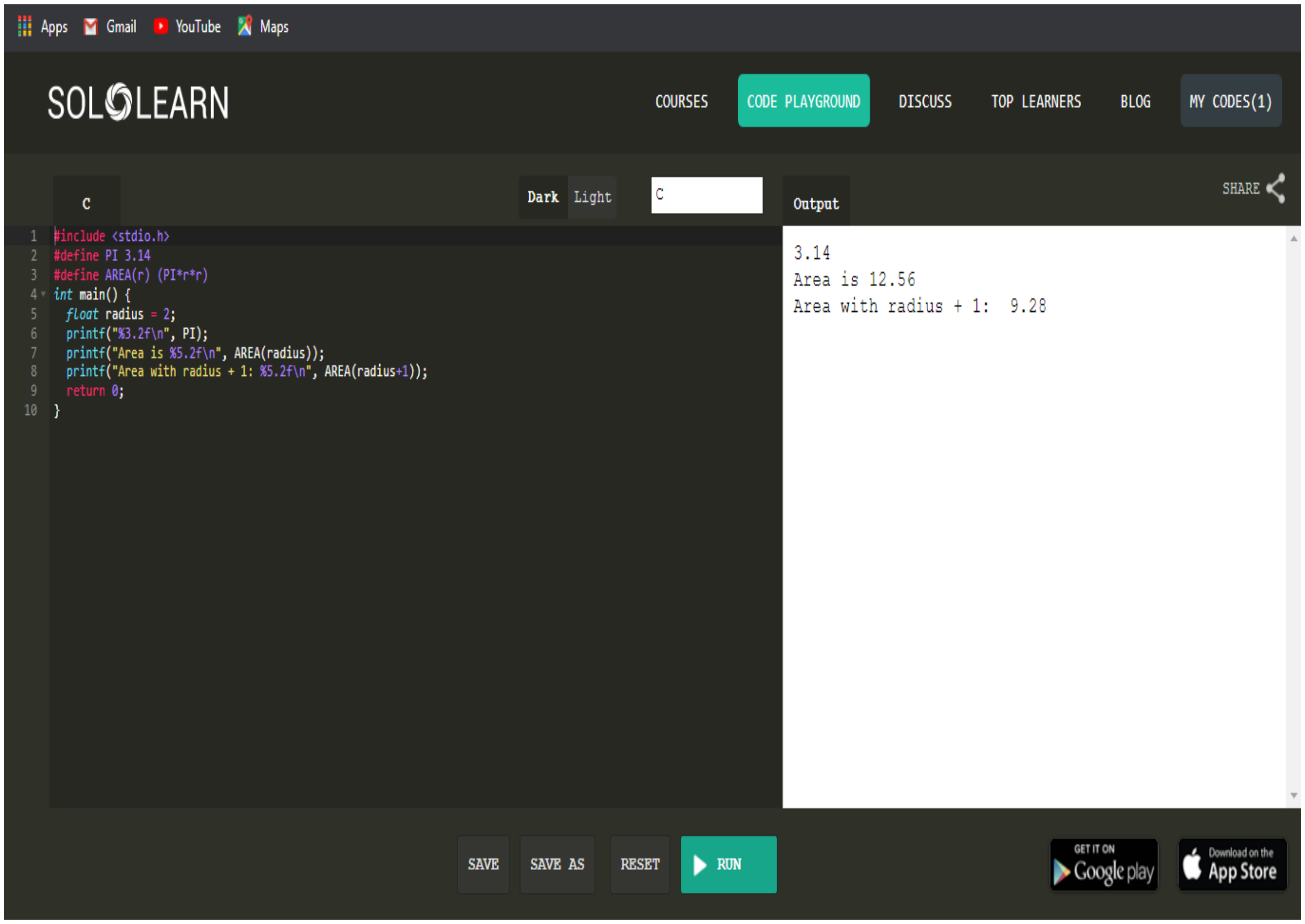
DAILY ASSESSMENT FORMAT

Date:	19/06/2020	Name:	Akshatha M Deshpande
Course:	SoloLearn	USN:	4AL17EC006
Topic:	C Programming	Semester & Section:	6th Sem A sec
Github Repository:	AkshathaDeshpande		

FORENOON SESSION DETAILS
Image of session



```c
#include <stdio.h>
#define PI 3.14
#define AREA(r) (PI*r*r)
int main() {
  float radius = 2;
  printf("%3.2f\n", PI);
  printf("Area is %5.2f\n", AREA(radius));
  printf("Area with radius + 1: %5.2f\n", AREA(radius+1));
  return 0;
}
```

```
3.14
Area is 12.56
Area with radius + 1:  9.28
```

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

Module 7:

Accessing Files:

- An external file can be opened, read from, and written to in a C program. For these operations, C includes the FILE type for defining a file stream. The file stream keeps track of where reading and writing last occurred.
- The stdio.h library includes file handling functions:
- FILE Typedef for defining a file pointer.
- fopen(filename, mode) Returns a FILE pointer to file filename which is opened using mode. If a file cannot be opened, NULL is returned.
- Mode options are:

  - r open for reading (file must exist)

  - w open for writing (file need not exist)

  - a open for append (file need not exist)

  - r+ open for reading and writing from beginning

  - w+ open for reading and writing, overwriting file

  - a+ open for reading and writing, appending to file

- fclose(fp) Closes file opened with FILE fp, returning 0 if close was successful. EOF (end of file) is returned if there is an error in closing.

Reading from a File:

- The stdio.h library also includes functions for reading from an open file.
- A file can be read one character at a time or an entire string can be read into a character buffer, which is typically a char array used for temporary storage.

Writing to a File:

- The stdio.h library also includes functions for writing to a file. When writing to a file, newline characters '\n' must be explicitly added.

Binary files:

- Binary file mode options for the fopen() function are:
- - rb open for reading (file must exist)
- - wb open for writing (file need not exist)
- - ab open for append (file need not exist)
- - rb+ open for reading and writing from beginning
- - wb+ open for reading and writing, overwriting file
- - ab+ open for reading and writing, appending to file

Controlling the File Pointer:

- There are functions in stdio.h for controlling the location of the file pointer in a binary file:
- ftell(fp) Returns a long int value corresponding to the fp file pointer position in number of bytes from the start of the file.

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.

The exit Command:

- Using exit to avoid a program crash is a good practice because it closes any open file connections and processes.

Using errno:

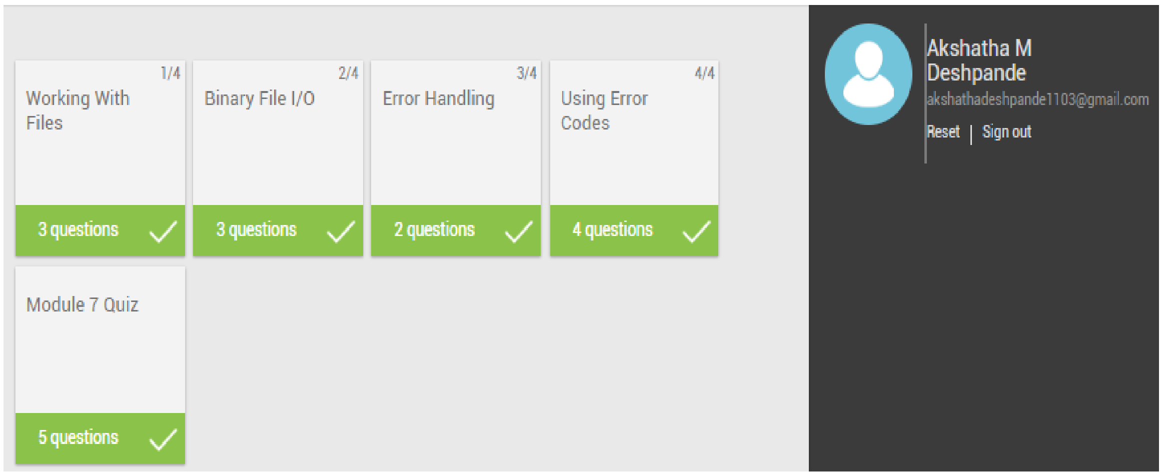
- To output the error code stored in errno, you use fprintf to print to the stderr file stream, the standard error output to the screen.
- Using stderr is a matter of convention and a good programming practice.

EDOM and ERANGE Error Codes:

- Some of the mathematical functions in the math.h library set errno to the defined macro value EDOM when a domain is out of range.

The feof and ferror Functions:

- In addition to checking for a NULL file pointer and using errno, the feof() and ferror() functions can be used for determining file I/O errors:
- feof(fp) Returns a nonzero value if the end of stream has been reached, 0 otherwise. feof also sets EOF.
- ferror(fp) Returns a nonzero value if there is an error, 0 for no error



MODULE 8:

Preprocessor Directives:

- The C preprocessor uses the # directives to make substitutions in program source code before compilation.
- For example, the line #include <stdio.h> is replaced by the contents of the stdio.h header file before a program is compiled.

The #include Directive:

- stdio input/output functions, including printf and file operations.
- stdlib memory management and other utilities
- string functions for handling strings
- errno errno global variable and error code macros

- math common mathematical functions
- time time/date utilities.

Formatting Preprocessor Directives:

- When using preprocessor directives, the # must be the first character on a line. But there can be any amount of white space before # and between the # and the directive.
- If a # directive is lengthy, you can use the \ continuation character to extend the definition over more than one line.

Predefined Macro Definitions:

- In addition to defining your own macros, there are several standard predefined macros that are always available in a C program without requiring the #define directive:
- __DATE__ The current date as a string in the format Mm dd yyyy
- __TIME__ The current time as a string in the format hh:mm:ss
- __FILE__ The current filename as a string
- __LINE__ The current line number as an int value
- __STDC__ 1

Conditional Compilation Directives:

- Conditional compilation of segments of code is controlled by a set of directives: #if, #else, #elif, and #endif.

Preprocessor Operators:

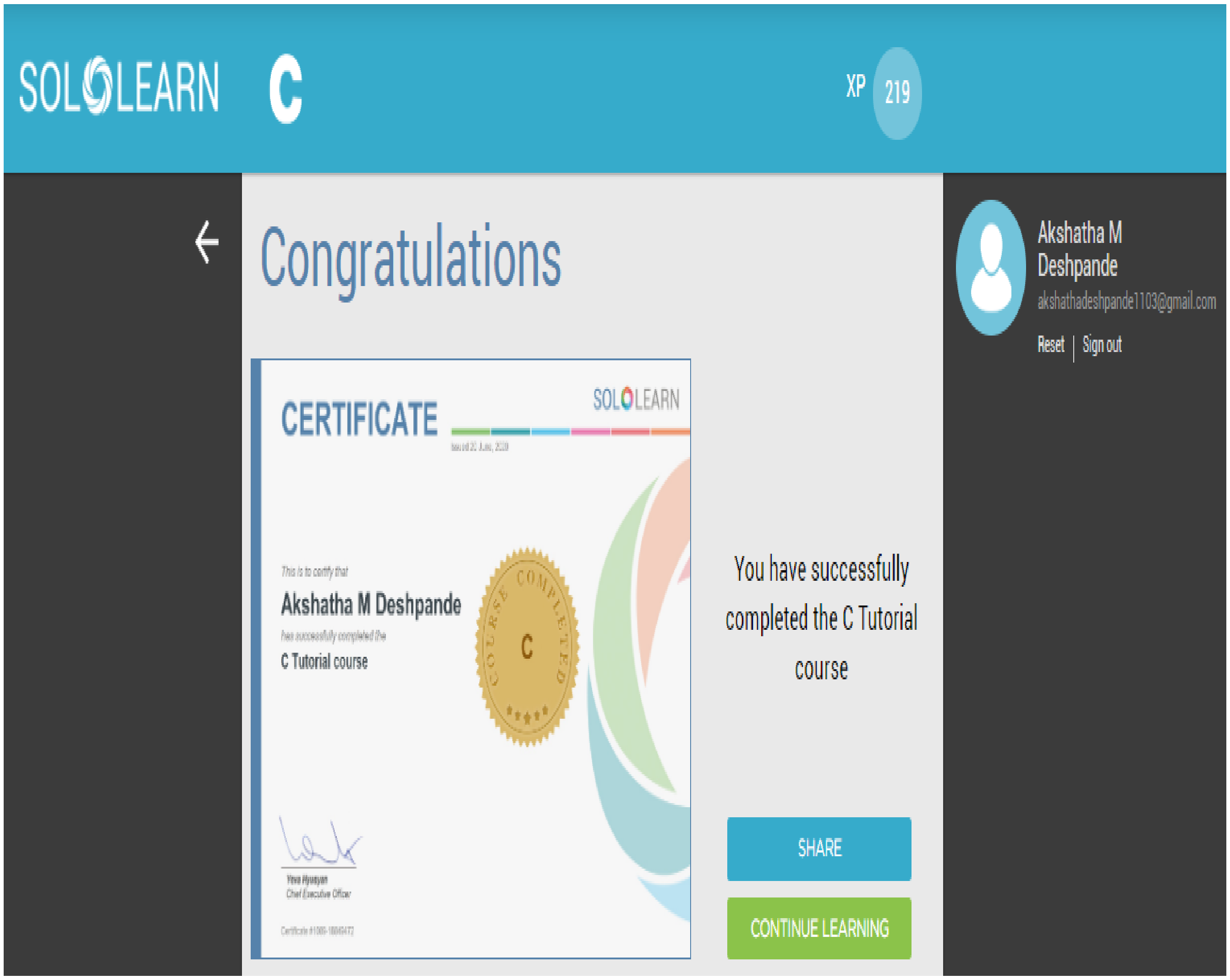
- The C preprocessor provides the following operators.
- The # Operator
- The # macro operator is called the stringification or stringizing operator and tells the preprocessor to convert a parameter to a string constant.
- White space on either side of the argument are ignored and escape sequences are recognized.

The ## Operator:

- The ## operator is also called the token pasting operator because it appends, or

  "pastes", tokens together.

CERTIFICATE:



Congratulations

CERTIFICATE

SOLOLEARN

Issued 20 June, 2020

This is to certify that

Akshatha M Deshpande

has successfully completed the

C Tutorial course

COURSE COMPLETED C

Yeva Hyusyan
Chief Executive Officer

Certificate #1089-18064472

You have successfully completed the C Tutorial course