Bibek Dhungana

Lab 9 - In Lab Assignment

Due end of the lab session

Acknowledge your collaborators or source of solutions, if any. **Submission by the end of the LAB is required.** Please type your answers, handwritten submission will not be accepted. Do all the following.

1. **In the programming language C, a_____file contains information about what a specific library's functions will do. The_____file contains the details of how these actions are executed.**
   .h file ---- contain information about what specific library will do.
   .c file ---- contain information about the details of how these actions are required.

   While writing c libraries, we need to define two files.
   a. .h file --- for declaring function prototype
   b. .c file ----for implementing those function definitions.
   For eg: stdio library consists of stdio.h containing all information what this library does. (function prototype and documentation) and stdio.c consists of all those implementations.
   Hence, stdio.c must be linked to program, which is automatically done by most complier or we need to do manually using -l for linking.

2. **When are variables of storage class static allocated? When are they deallocated?**
   Static variables preserve its value even when the variable is out of scope. So, the scope of static variable is the entire file.
   Hence, static variables are allocated in data segment when your program is loaded. And the static variable is deallocated at the end of the program.

3. **What will the following code print? Please explain the behaviour of the code in a few words. (What is it printing and why is it printing that?**

```
#include <stdio.h>
#define ISEQUAL(X, Y) X == Y
int main()
{
	#if ISEQUAL(X, 0)
		printf("x");
	#else
		printf("y");
	#endif
	return 0;
}
```

The output of the function is:
x.
This code is simply replacing ISEQUAL(X,0) with X == Y as defined in macro
`#define ISEQUAL (X, Y) X == Y`
. X is declared to be 0. Since, 0 == 0 is 1(true). We get x as an output.

This replacement is done by preprocessor. So, we can get see what we get as after the code is acted by preprocessor by using -E flag on gcc compliler.

gcc -E main.c
we get,
int main ()
{

printf("x");

return 0;
}

Hence, the output is x.