Lab 2

Basics and Printing

Learning Objectives

After completing this lab you will be able to do some basic printing and simple math in C.

Basic Lab Instructions!

- Talk to your classmates for help.
- You may want to bring your textbook to future labs to look up syntax and examples.
- Stuck? Confused? Have a question? Ask a TA/Lab Engineer for help, or look at the book or past lecture slides.
- Complete as many problems as you can within the allotted time. You don't need to keep working on these exercises after you leave the lab.
- Before you leave today, make sure to check in with one of the Lab Engineers/TAs in the lab to get credit for your work.

Lab Tasks

Creating a New Project

After completing this section you will have created a project for today's lab.

1. Open the solution CS110Labs you created in the lab last week.
2. Add a new project called Lab01 within the solution CS110Labs. Stop here! start with the lab tasks, and refer to the following steps accordingly.
3. Click File in the top menu bar of Visual Studio, and then select the Add New Item option (Add Existing Item option depending on the lab task).
4. Select C++ File
5. Type in the code listed.
6. Compile (Build Lab01) the program and execute it (Execute without Debugging).

Task #1: Comments and Slashes

Comments are very important in your programs. They are used to tell you what something does in English, and they also are used to disable parts of your program if you need to remove them temporarily. Here's how you use comments in C:

```c
#include <stdlib.h>
#include <stdio.h>

int main()
{
  // A comment, this is so you can read your program later.
  // Anything after the // is ignored by C/C++/Java.

  printf( "I could have code like this.\n" ); // and the comment after is ignored.

  /* Want to use C-style to comment */

  /* You can even write comments
  spanning multiple lines
  using the C-style */

  // You can also use a comment to "disable" or comment out a piece of code:
  // printf( "This won't run.\n" );

  printf( "This will run.\n" );

  return EXIT_SUCCESS;
}
```

The code is provided to you as CommentsAndSlashes.c. Compile and execute it.

What You Should See

```
I could have code like this.
This will run.
```

What You Should Do on Your Own

For this task, try these changes.

1. Were you right about what the two slashes ('//') signify? Answer in a comment at the top of the file (above the '#include' lines).
2. Add another comment at the very top of the file (above your answer to the previous question) with your name and today's date.

You have turn-in the modified CommentsAndSlashes.c file. Remember assignments turned in without the above changes will not receive any points.

Task #2: Indentation

Programs should be indented properly to make them easier to read:

- { brace ← increase indent of following lines by one tab
- } brace ← decrease indent of that line and following lines by one tab

Example.

```c
#include <stdlib.h>
#include <stdio.h>

int main()
{
  printf( "Hello, world!\n" );
  printf( "How are you?\n" );

  return EXIT_SUCCESS;
}
```

Make sure that from this day on all your program has good indentation.

What You Should Do on Your Own

For this task, open CodeAndIndent.c in your solution and try these.

1. Correctly indent the code.
2. How many lines of output are produced (including blank lines)?

You have turn-in the modified CodeAndIndent.c file.

Task #3: Escape Sequences

An escape sequence inserts a special character into a printf statement.

Sequence	Special character
\n	new line (goes to the next line)
\t	tab (indents output by roughly 8 spaces)
\"	quotation mark
\\	backslash

Example.

```c
  printf( "I said \"hello\" to Areeb.\n" );
```

What You Should See

```
I said "hello" to Areeb.
```

What You Should Do on Your Own

For this task, try these.

1. Open EscapeSequences.c. What output is produced by the code?

   ```c
     printf( "C is fun!\n");
     printf( "The string \"\" is an empty message.\n");
     printf( "\\'\"\\\\\"\n");
   ```

   Try to figure it out without running the code.

2. Write a complete C program named MuchBetter that produces the following output (note the blank line):

```
A "quoted" String is
'much' better if you learn
the rules of "escape sequences."

Also, "" represents an empty String.
Don't forget: use \" instead of " !
'' is not the same as "
```

You have turn-in the CodeAndIndent.c and MuchBetter.c file.

Task #4: A Letter to Yourself

Write a program that displays your name and address on the screen as if it were a letter. Your output should look roughly something like that below.

```
+---------------------------------------------+
| Pakistan Post                       ####### |
|                                     ####### |
|             Anis Rahman             ####### |
|             IAEC 204, NUST-SEECS    ####### |
|             Islamabad, Pakistan             |
|                  _ _ _ _ _                  |
|                 |_|_|_|_|_|                 |
|                 Postal Code                 |
+---------------------------------------------+
```

You have turn-in the LetterToYourself.c file.

Task #5: Your Initials

Display your initials on the screen in block letters as shown.

```
For the name Anis Rahman ...

  A     RRRR
 A A    R   R
A   A   R   R
AAAAA   RRRR
A   A   R R
A   A   R  R
A   A   R   R
```

You have turn-in the YourInitials.c file.

Task #6: Numbers and Math

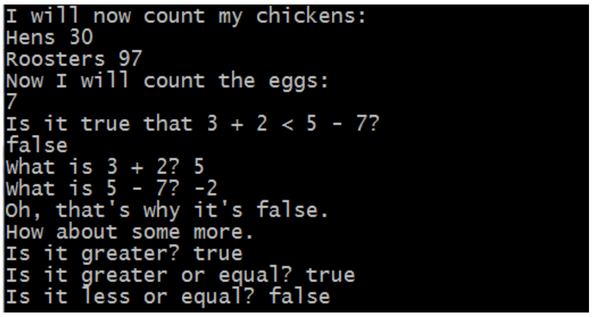
Every programming language has some kind of way of doing numbers and math. This exercise has lots of math symbols so let's name them right away so you know what they're called. Here are the names:

Symbol	Name	Symbol	Name
+	plus	<	less-than
-	minus	>	greater-than
/	slash	<=	less-than-or-equal
*	asterisk	>=	greater-than-or-equal
%	percent		

Open NumbersAndMath.c in you solution. Compile and execute it.

```c
#include <stdlib.h>
#include <stdio.h>

int main()
{
  printf( "I will now count my chickens:\n" );

  printf( "Hens %d\n", ( 25 + 30 / 6 ) );
  printf( "Roosters %d\n", ( 100 - 25 * 3 % 4 ) );

  printf( "Now I will count the eggs:\n" );

  printf( "%d\n", 3 + 2 + 1 - 5 + 4 % 2 - 1 / 4 + 6 );

  printf( "Is it true that 3 + 2 < 5 - 7?\n" );

  printf( "%s\n", ( ( 3 + 2 < 5 - 7 ) ? "true" : "false" ) );

  printf( "What is 3 + 2? %d\n", ( 3 + 2 ) );
  printf( "What is 5 - 7? %d\n", ( 5 - 7 ) );

  printf( "Oh, that's why it's false.\n" );

  printf( "How about some more.\n" );

  printf( "Is it greater? %s\n", ( ( 5 > -2 ) ? "true" : "false") );
  printf( "Is it greater or equal? %s\n", ( ( 5 >= -2 ) ? "true" : "false") );
  printf( "Is it less or equal? %s\n", ( ( 5 <= -2 ) ? "true" : "false") );

  return EXIT_SUCCESS;
}
```

What You Should See



```
I will now count my chickens:
Hens 30
Roosters 97
Now I will count the eggs:
7
Is it true that 3 + 2 < 5 - 7?
false
What is 3 + 2? 5
What is 5 - 7? -2
Oh, that's why it's false.
How about some more.
Is it greater? true
Is it greater or equal? true
Is it less or equal? false
```

What You Should Do on Your Own

Assignments turned in without these things will not receive any points.

1. Above each line, use two slashes // to write a comment to yourself explaining what the line does.
2. Notice the math seems "wrong"? There are no fractions, only whole numbers. Find out why by researching what a "floating point" number is.
3. Rewrite NumbersAndMath.c to use floating point numbers so it’s more accurate (hint: 20.0 is floating point).

You have turn-in the modified NumbersAndMath.c file.

Optional: Bonus Challenge

Write a C program Dragon.c to print out a Dragon using just letters, digits, and punctuation symbols.

Your first attempt might be,

```c
#include <stdlib.h>
#include <stdio.h>

int main()
{
  printf( "+----+__0\n");
  printf( "|    |\n");
  printf( "+----+\n");
  printf( "|    |" );

  return EXIT_SUCCESS;
}
```

The output in concole might look like,

```
 +----+__0
 |    |
 +----+
 |    |
```

Experiment, share and improve on it. At the end of this lab your can claim to have produced the majestic dragon.

Hand in

Hand in the source code from this lab at the appropriate location on the blackboard system at LMS. You should hand in a single compressed/archived file named Lab_2_<your reg. No. XXX without angle brackets>.zip that contains the following.

1. All completed C source files representing the work accomplished for this lab: CommentsAndSlashes.c; EscapeSequences.c; CodeAndIndent.c; MuchBetter.c; LetterToYourself.c; YourInitials.c; NumbersAndMath.c; and Dragon.c. The files should contain author in the comments at the top.

2. An plain text file named OUTPUT.txt that includes a) author information at the beginning, b) a brief explanation of the lab, and c) any comments, or suggestions.

To Receive Credit

1. By showing up on time for lab, working on the lab solution, and staying to the end of the class period, only then you can receive full credit for the lab assignment.

2. Comment your program heavily. Intelligent comments and a clean, readable formatting of your code account for 20% of your grade.

3. In-class lab time is not intended as free time for working on your program assignments. Only if you have completely solved the lab assignment, including all challenges, and have had your work checked off for completeness by your TA/Lab Engineer should you begin the program assignment.