Lab 6

While Loops

Learning Objectives

Want to repeat some instructions multiple times, you need to use repitition. C gives you several constructs suitable for such scenarios. Lets start with the while loops that repeatedly executes a set of instructions until the conditions is false, or when it fails.

After completing this lab you will be able to write repitition statements that cause a section of code to be executed multiple times depending on a conditional.

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 Lab05 within the solution CS110Labs. Set the project as startup project (right-click the project in Solution Explorer). 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 Lab05) the program and execute it (Execute without Debugging).

Task #1: Enter Your PIN

Type in the following code (EnterPIN.c), and get it to compile. This assignment will help you learn how to make a loop, so that you can repeat a section of code over and over again!

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

int main()
{
  int pin = 1234, entry;

  printf( "WELCOME TO THE SEECS BANK.\n" );
  printf( "ENTER YOUR PIN: " );
  scanf( "%d", &entry );

  while ( entry != pin )
  {
    printf( "\nINCORRECT PIN. TRY AGAIN.\n" );
    printf( "ENTER YOUR PIN: " );
    scanf( "%d", &entry );
  }

  printf("\nPIN ACCEPTED. YOU NOW HAVE ACCESS TO YOUR ACCOUNT.");

  return EXIT_SUCCESS;
}
```

What You Should See

Sample run 1.

```
WELCOME TO THE SEECS BANK.
ENTER YOUR PIN: 9021

INCORRECT PIN. TRY AGAIN.
ENTER YOUR PIN: 1111

INCORRECT PIN. TRY AGAIN.
ENTER YOUR PIN: 1234

PIN ACCEPTED. YOU NOW HAVE ACCESS TO YOUR ACCOUNT.
```

Sample run 2.

```
WELCOME TO THE SEECS BANK.
ENTER YOUR PIN: 1234

PIN ACCEPTED. YOU NOW HAVE ACCESS TO YOUR ACCOUNT.
```

What You Should Do on Your Own

Assignments turned in without these things will receive no credit.

1. How is a while loop similar to an if statement?
2. How is a while loop different from an if statement?
3. Delete the scanf statement from inside the while loop. What happens? Why? (Put the line back before you turn in the assignment.)

Task #2: A Number-Guessing Game

Write a program that plays an incredibly stupid number-guessing game. The user will try to guess the secret number until they get it right. That means it will keep looping as long as the guess is different from the secret number. Use a while loop.

You must store the secret number in a variable, and use that variable throughout. The secret number itself must not appear in the program at all, except in the one line where you store it into a variable.

```
I have chosen a number between 1 and 10. Try to guess it.
Your guess: 5
That is incorrect. Guess again.
Your guess: 4
That is incorrect. Guess again.
Your guess: 8
That is incorrect. Guess again.
Your guess: 6
That's right! You guessed it.
```

Task #3: Dice Doubles

Write a program DiceDoubles.c that simulates a dice roll by picking a random number from 1-6 and then picking a second random number from 1-6. Add the two values together, and display the total. The dice game keeps rolling until they get doubles (the same number on both dice).

```
HERE COMES THE DICE!

Roll #1: 3
Roll #2: 5
The total is 8!

Roll #1: 6
Roll #2: 1
The total is 7!

Roll #1: 2
Roll #2: 5
The total is 7!

Roll #1: 1
Roll #2: 1
The total is 2!
```

Task #4: Counting with a While Loop

Type in the following code (CountingWhile.c), and get it to compile. This assignment shows you how we can use a while loop to make something repeat an exact number of times.

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

int main( )
{
  char message[80];

  printf( "Type in a message, and I'll display it five times.\n" );
  printf( "Message: " );
  gets(message);

  int n = 0;
```

```
13    while ( n < 5 )
14    {
15      printf( "%d. %s\n", (n+1), message );
16      n++;
17    }
18
19    return EXIT_SUCCESS;
20  }
```

Normally, while loops are best for repeating as long as something is true:

- ❖ Keep going as long as they haven't guessed it.
- ❖ Keep going as long as you haven't got doubles.
- ❖ Keep going as long as they keep typing in a negative number.
- ❖ Keep going as long as they haven't typed in a zero.

But sometimes, we know in advance how many times we want to do something.

- ❖ Do this ten times.
- ❖ Do this five times.
- ❖ Pick a random number, and do it that many times.
- ❖ Take this list of items, and do it one time for each item in the list.

We can do that sort of thing with a while loop, but we have to use a counter. A counter is a number variable (int or double) that starts with a value of 0, and then we add 1 to it whenever something happens. So, here, we're going to be adding 1 to the counter everytime we repeat the loop. And when the counter reaches a predetermined value, we'll stop looping.

What You Should See

```
1  Type in a message, and I'll display it five times.
2
3  Message: The only source of knowledge is experience.
4  1. The only source of knowledge is experience.
5  2. The only source of knowledge is experience.
6  3. The only source of knowledge is experience.
7  4. The only source of knowledge is experience.
8  5. The only source of knowledge is experience.
```

What You Should Do on Your Own

1. What does n++ do? Remove it and see what happens. (Then put it back.)
2. Change the code so that the loop repeats ten times instead of five.
3. See if you can change the code so that the message still prints ten times, but the numbers in front count by tens, like so:

```
1  Type in a message, and I'll display it ten times.
2
3  Message: The only source of knowledge is experience.
4  10. The only source of knowledge is experience.
5  20. The only source of knowledge is experience.
```

```
30. The only source of knowledge is experience.
40. The only source of knowledge is experience.
50. The only source of knowledge is experience.
60. The only source of knowledge is experience.
70. The only source of knowledge is experience.
80. The only source of knowledge is experience.
90. The only source of knowledge is experience.
100. The only source of knowledge is experience.
```

4. Change the code so that it asks the person how many times to display the message. Then, print it that many times. Still count by tens.

```
Type in a message, and I'll display it several times.

Message: The only source of knowledge is experience.
How many times? 3
10. The only source of knowledge is experience.
20. The only source of knowledge is experience.
30. The only source of knowledge is experience.
```

Task #5: PIN Lockout

Type in the following code (PinLockout.c), and get it to compile.

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

int main( )
{
  int pin = 1234;
  int tries = 0;
  int entry = 0;

  printf( "WELCOME TO THE SEECS BANK.\n" );

  printf("\nENTER YOUR PIN: ");
  scanf( " %d", &entry );
  tries++;

  while ( entry != pin && tries < 3 )
  {
    printf("\nINCORRECT PIN. TRY AGAIN.\n");

    printf("ENTER YOUR PIN: ");
    scanf( " %d", &entry );
    tries++;
  }

  if ( entry == pin )
    printf("\nPIN ACCEPTED. YOU NOW HAVE ACCESS TO YOUR ACCOUNT.\n");
  else if ( tries >= 3 )
    printf("\nYOU HAVE RUN OUT OF TRIES. ACCOUNT LOCKED.\n");

  return EXIT_SUCCESS;
}
```

What You Should See

```
WELCOME TO THE SEECS BANK.
ENTER YOUR PIN: 1010

INCORRECT PIN. TRY AGAIN.
ENTER YOUR PIN: 2323

```

```
INCORRECT PIN. TRY AGAIN.
ENTER YOUR PIN: 9999

YOU HAVE RUN OUT OF TRIES. ACCOUNT LOCKED.
```

What You Should Do on Your Own

1. Change the code so that it locks them out after 4 tries instead of 3. Make sure to change the message at the bottom, too.

2. Move the "maximum tries" value into a variable, and use that variable everywhere instead of just the number.

Task #6: Number-Guessing with a Counter

Modify your previous number-guessing game to NumberGuessingWithACounter.c so that they can guess until they get it right, and count the number of tries it takes them to guess it.

```
I have chosen a number between 1 and 10. Try to guess it.
Your guess: 5
That is incorrect.  Guess again.
Your guess: 4
That is incorrect.  Guess again.
Your guess: 8
That is incorrect.  Guess again.
Your guess: 6
That's right! You guessed it.
It only took you 4 tries.
```

Task #7: Hi-Lo with Limited Tries

Write a program HiLoLimited.c that picks a random number from 1-100. The user keeps guessing as long as their guess is wrong, and they've guessed less than 7 times. If their guess is higher than the number, say "Too high." If their guess is lower than the number, say "Too low." When they get it right, the game stops. Or, if they hit seven guesses, the game stops even if they never got it right.

This means your while loop will have a compound condition using &&.

Sample run 1.

```
I'm thinking of a number between 1-100.  You have 7 guesses.
First guess: 50
Sorry, you are too low.
Guess # 2: 75
Sorry, you are too low.
Guess # 3: 87
Sorry, that guess is too high.
Guess # 4: 82
Sorry, you are too low.
Guess # 5: 84
You guessed it!
```

Sample run 2.

```
I'm thinking of a number between 1-100.  You have 7 guesses.
First guess: 50
Sorry, you are too low.
Guess # 2: 75
Sorry, you are too low.
Guess # 3: 87
```

```
Sorry, that guess is too high.
Guess # 4: 82
Sorry, you are too low.
Guess # 5: 84
You guessed it!
```

Task #8: Adding Values in a Loop

Write a program AddingValuesInALoop.c that gets several integers from the user. Sum up all the integers they give you. Stop looping when they enter a 0. Display the total at the end.

You must use a while loop.

```
I will add up the numbers you give me.
Number: 6
The total so far is 6
Number: 9
The total so far is 15
Number: -3
The total so far is 12
Number: 2
The total so far is 14
Number: 0

The total is 14.
```

Hand in

Hand in the word file for this lab at the appropriate location on the blackboard system at LMS. You should hand in a single file named Lab_6_<your reg. No. XXX without angle brackets>.doc(x) that contains the following.

1. All completed C source files representing the work accomplished for this lab: EnterPIN.c; KeepGuessing.c; DiceDoubles.c; CountingWhile.c; PinLockout.c; NumberGuessingWithACounter.c; HiLoLimited.c; and AddingValuesInALoop.c.
2. A paragraph at the end that includes a) a brief explanation of the lab, and b) 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.