

Week of March 8, 2015

Topics for this week: Flow of Control: Loops

Activity Checklist

	Read chapter 10 in your course packet.
	Review the slides Solving Problems with Repetition
	Review the sample program Loops
	Review the video Loops
	Complete lab #18 , due by 11:59pm on Thursday.
	Complete lab #19 , due by 11:59pm on Thursday.
	Complete Project #7 and submit it to Canvas before 11:59pm on Sunday. Late programs will lose 20% of the possible points for each day that they are late. If you turn this program in prior to 11:59pm on Saturday you will receive a 5 point bonus, if it meets all of the specifications and gives the correct answers.

Learning Goals

It is expected that you will meet the objectives outlined here by the end of the week. You might want to test yourself to see how well you fare. You can be guaranteed that you will be tested on these concepts on your next exam. By the end of this unit, you should be able to:

- Explain how the while statement works and correctly use it in a C# program.
- Explain how the do-while statement works and correctly use it in a C# program.
- Explain how the for statement works and correctly use it in a C# program.
- Correctly use nested loops in a C# program.
- Use loops to validate user input.
- Design simple algorithms that use branches and loops.
- Diagram simple algorithms using an activity diagram.
- Create programs that use random numbers.
- Test an algorithm.

Reading Assignment

All reading should be done before you come to class. Your ability to understand the material discussed in class will be greatly enhanced when you come to class prepared.

1. Chapter 9 in the course packet discusses problems that involve repetition.
2. The slides for this week discuss **for loops**, **while loops**, and **do-while loops**.

Key Concepts

Here are some important concepts from this unit that you should be sure you understand.

1. Loops provide a mechanism for repeating blocks of code until some condition occurs. In C# we study four different looping mechanisms: the **while loop**, the **do-while loop**, and the **for loop**.
2. A *do-while* loop is guaranteed to always execute the body of the loop one time. There are situations where the *while* loop and *for* loop might not execute the body of the loop at all.
3. A *for* loop is best used when you know that you need to iterate through the loop a given number of times.
The best time to use a *while* or *do-while* loop is when the condition controlling the loop changes within the body of the loop.
4. The C# language provides the *break* and *continue* statements to alter the normal operation of a loop. Use these statements sparingly!

Lab Assignment

This week you should complete labs 19 and 20. These labs explain how to use loops to validate input, to test for yes and no answers, and how to generate random numbers in your programs. Be sure to read all of the study material associated with these labs before you try to complete the lab projects.

Programming Project

This week you should complete your project #7. It will test your abilities to design, code and test a C# program using looping logic. It will also introduce you to Fibonacci numbers.