Repetition

CS 133N/ CS 161N

Mari Good

Objectives

- Review programming control structures
- Introduce algorithms and pseudocode involving Repetition
- Introduce you to C# syntax for
  - while loop
  - do while loop
  - for loop
- Practice with several examples

Control Structures

- All programming languages have 3 control structures. These are sometimes referred to as structured programming constructs. They are
  - Sequence – in the absence of any other construct, programming statements are executed in order. All of your programs thus far have used sequence.
  - Selection – allows programs to branch or make choices based on a condition. That was the focus of the last topic
  - Repetition – allows programs to execute blocks of code 1 or more times based on a condition. That's the focus of this topic.

Let's start with an example

- Design and implement a program that asks the user to enter his/her/their name and his/her/their favorite number. The program should then display a welcome message to the user using his/her/their name repeatedly.
  - The first step is to understand the problem. Questions?
  - The second step is to use an IPO chart to describe WHAT needs to be done, in a very general way, to solve the problem.
    - It's still important to identify the input and output but I'll skip the processing steps from now on and write an algorithm in pseudocode instead.

IPO Chart

Input
- name
- favoriteNumber

Processing
- We'll write an algorithm instead

Output
- name – as many times as favoriteNumber

Algorithms and Pseudocode

- Here's an algorithm for the problem

```
display instructions
get name
get favoriteNumber
count = 1
while count <= favoriteNumber
        display welcome + name
        count = count + 1
end while
```

This repetition structure allow the code to execute multiple times. The while loop is one repetition statement. It starts with while and ends with end while

When the condition is true the body of the loop executes and the condition is reevaluated. When the condition is false the loop stops executing.

Testing an Algorithm

```
display instructions
get name
get favoriteNumber
count = 1
while count <= favoriteNumber
        display welcome + name
        count = count + 1
end while
```

name	favoriteNumber	count

Another Example

- Design and implement a program that asks the user to enter a number and counts down from that number.
  - The first step is to understand the problem. Questions?
  - The second step is to use an IPO chart to describe WHAT needs to be done, in a very general way, to solve the problem.
    - It's still important to identify the input and output but I'll skip the processing steps from now on and write an algorithm in pseudocode instead.

IPO Chart

- Input
  - number
- Processing
  - We'll write an algorithm instead
- Output
  - number
  - number – 1
  - number – 2
  - … until number = 1

Algorithms and Pseudocode

- Let's see if we can write an algorithm

```
display instructions
get number
while number >= 1
    display number
    number = number - 1
end while
```

Pretest loop.

I could have written this with a do while loop. Notice that the condition is at the bottom … so the body executes at least once

```
do
    display number
    number = number -1
while number >= 1
```

Posttest loop

Testing an Algorithm

```
display instructions
get number
while number >= 1
    display number
    number = number - 1
end while
```

number	display

Our Third Example

- Design and implement a loop that validates input from the user. The user should be prompted to enter a number between 1 and 10 and should be forced to continue entering numbers until the number he/she/they enters is in the correct range.
  - The first step is to understand the problem. Questions?
  - The second step is to use an IPO chart to describe WHAT needs to be done, in a very general way, to solve the problem.
    - It's still important to identify the input and output but I'll skip the processing steps from now on and write an algorithm in pseudocode instead.

Algorithms and Pseudocode

- Let's see if we can write an algorithm

```
display instructions
get number
while number < 1 or number > 10
    ask user to re-enter
    get number
end while
```

With a do while loop.

```
do
    ask user to enter
    get number
while number <1 or number > 10
```

Your Turn

- Let's look at the 6 problems that are part of lab 4 together. For each problems you should
  - Do the I and O part of the IPO chart
  - Do one or more examples
  - Write an algorithm in pseudocode
  - Test the algorithm
- Next time I'll show you how to translate the loops into C# code

Translating an Algorithm into Code

- Each line in your pseudocode will be replaced with one or more statements of C# code.
- You already know how to
  - Declare variables
  - Get input
  - Assign values to variables
  - Write if statements
  - Display output
- The only new syntax is the loops. There are 3 in C#.

General Syntax for while

Conditions use relational and logical operators just like in an if statement.

```
while (condition)
{
    statements that execute when condition is true
}
```

{ } can be omitted if there's only one statement in the block

General Syntax for do while

```
do
{
    statements that execute when condition is true
}
while (condition);
```

Notice the ; at the end of the statement (condition). This is the only kind of loop (or selection statement) what will have a ; after the condition!

General Syntax for for

Executes once.

Is evaluated every time before the loop body is executed

Is evaluated every time after the loop body and before the condition is evaluated

```
for (initialization; condition; update)
{
    statements that execute when condition is true
}
```

Example:

```
for (int count = 1; count <= favoriteNumber; count++)
    Console.WriteLine("Hello World");
```

More examples

- Let's look at the first 3 examples we used for practicing pseudocode in dotnetfiddle.net.
- Then we'll use dotnetfiddle.net to do the first 2 problems from the lab together.
- Finally, there will be time to do the other 4 problems from the lab (you've already done the IPO charts and algorithms) in small groups. I'll help whenever you get stuck.

What's Next

- Midterm
  - Practice quizzes will help you get ready.
  - Midterm is in class. If you want to take it outside of the classroom or at another time, you must make arrangements with me in advance.
- Don't forget
  - Reading Quiz 4
  - Programming Quiz 4
  - Lab 4 – 6 problems