Chapter 5
Loops

FORDHAM

CISC 5004 - Programming C++
Sam Kamens
Fall, 2022

Loops (General)

- Loop Concept
- Loop Basics
  - Loop Body: Statements inside the loop
  - Condition: Expression that is evaluated to determine whether or not to exit the loop
  - Iteration: One time through the loop
- Examples
  - Sum of Input Values
  - Computing an Average

While Loops

```
while (expression) { // Loop expression
    // Loop body: Executes if expression evaluated to true
    // After body, execution jumps back to the "while"
}
// Statements that execute after the expression evaluates to
// false
```

- Example

While Loops

- General Pattern
  - Set initial value for while condition
    - May be obtained using user input
  - Execute loop statements
  - At end of loop, obtain next value for while condition

```
// Get input into userChar

while (userChar == 'y') {
   // Do something ...
   // Get input into userChar
}
```

Loop expressions (or conditions)

- The loop condition can be any expression that evaluates to `true` or `false`
  - while (userChar == 'y')
  - while (userNum > 0)
  - while (consYear >= userYear)
- Example: Ancestor printing program

While Loops: Common errors

- Opposite loop expression
  - e.g. `x == 0` rather than `x != 0`
  - Expression describes when the loop *should* iterate, not when it should terminate
- Infinite Loop
  - A loop that never stops iterating
  - Examples

While Loops: Additional Examples

- Greatest Common Divisor
- Sentinel Value

do-while Loops

- Body is executed first, then the loop condition is checked
- ```
  do {
      // Loop body
  } while (loopExpression);
  ```
- Used when the loop should always iterate at least once
- Example
  ```

For Loops: Basics

```
int i;

i = 0;
while (i < 10) {
  cout << i << " ";
  ++i;
}
cout << endl;
```

```
int i;
for (i = 0; i < 10; ++i) {
  cout << i << " ";
}
cout << endl;
```

Increment/Decrement Operators

- Increment
  - `++i, --i` (pre-increment/decrement): Changes value before evaluation
  - `i++, i--` (post-increment/decrement): Changes value after evaluation
- `i = 5; j = i++; // j is 5, i is 6`
- `i = 5; j = ++i; // j is 6, i is 6`
- In general, pre- is more efficient than post- when not using the value

For Loop Examples

- Savings Interest
- Average of a list of values

Which loop type to use?

- `for`
  - Number of iterations can be computed before the loop starts
- `while`
  - Number of iterations can not easily be known in advance
- Pro tip: Any loop can be written either way; often it depends on readibility

For Loops: More examples

- Find the max in a list
- Incrementing by a number besides 1
- Celsius to Fahrenheit table

Loop Style Guidelines

- Iterate starting with 0
- Use pre-increment/decrement
- In-loop variable declaration
  - `for (int i = 0; i < N; ++i)`

Common Errors/Good Practice (?)

- Extra Increment
- Good practices (?)

Loops and Strings

- Iterate through a string
- Iterating until done

Nested Loops

- A loop inside the body of another loop
  - Inner loop vs. outer loop
- Two letter domain names
- Histogram

Developing Programs Incrementally

- Phone Number Example

Break and continue

- `break`
  - Immediately exit a loop
  - May simplify the code
    - But be careful about assumptions re: what code executes!
  - Example: Meal Finder
- `continue`
  - Go back to the condition check immediately
  - May improve readability
  - Example: Meal finder 2

Variable Name Scope

- Name is only valid within its **scope**
- In C++, **scope** is defined by the **block** in which the variable is declared
- Common problem: variable in the wrong scope
- For loop index
- Common error: declaring variable inside a loop block

Enumerations

- Enumeration Type (`enum`)
  - Stores only a small set of named values in a variable

```
enum identifier {enumerator1, enumerator2,  ...};
```

- Example
- Normally declared with values
  - `enum Months { January = 0, February = 1, March = 2, ... }`

Example Programs/Labs

- Salary Calculation
- Domain Name Validation
- Varied Amount of Input Data
- Remove non-alpha characters
- Print string in reverse