5.2 The while Loop

Friday, February 3, 2023 10:51 AM

- while loop executes statements repeatedly while condition is true
  - while(loop-continuation-condition){
  - //loop body
  - Statement(s);
  - }
- Pt of loop that repeated called loop body
- One-time execution called iteration (aka repetition)
- Loop-continuation-condition is bool, ctrl execution of body, evaled each time to see if loop body executed
- True=>executed
- Counter-controlled loop ctrls number of loop, usually increment/decrement var

5.3 Case Study: Guessing Numbers

Sunday, February 5, 2023 3:27 PM

- U guess numb in computer mind, it has random digit btwn 0 and 100 (inclusive), pgrm prompts user to enter numb continuously until match
- Each input will output correct or too high or too low
- To guess fast, guess half, then half, then half, …
- Code drafts or something for loops
-

```
#include <iostream>
#include <cstdlib>
#include <ctime> // Needed for the time function
using namespace std;

int main()
{
  // Generate a random number to be guessed
  srand(time(0));
  int number = rand() % 101;

  cout << "Guess a magic number between 0 and 100";

  int guess = -1;
  while (guess != number)
  {
    // Prompt the user to guess the number
    cout << "\nEnter your guess: ";
    cin >> guess;

    if (guess == number)
      cout << "Yes, the number is " << number << endl;
    else if (guess > number)
      cout << "Your guess is too high" << endl;
    else
      cout << "Your guess is too low" << endl;
  } // End of loop

  return 0;
}
```

-
-

5.4 Loop Design Strats

Sunday, February 5, 2023 3:49 PM

- Loop steps
    1. Id the statements need to repeat
    2. Wrap statements in loop
    3. Code condition and add the statements for loop ctrl
-

5.5 Controlling a Loop with User Confirmation or a Sentinel Value

Sunday, February 5, 2023 4:34 PM

- Check if user wants loop to continue by condition being something like continueLoop = 'Y' and then ask for their char.
- Also can do sentinel value, which is special val, loops by this called sentinel-ctrled loop
- Don't use float vals for equality checking in loop ctrled expression
-

5.6 Input and Output Redirections and Read All Data from a File

Sunday, February 5, 2023 5:12 PM

- Input redirection where pgrm takes input from file (.txt) instead of keyboard at runtime, "changes console input from a file"

The program will report that "The sum is 518". Note that **SentinelValue.exe** can be obtained using a Microsoft C++ command line compiler (See **Supplement Part I.B**):

```
cl SentinelValue.cpp
```

To compile it using a GNU C++ compiler, use the following command (See **Supplement Part I.B**):

```
g++ SentinelValue.cpp -o SentinelValue.exe
```

- Output redirection can send output to a file instead of to console

5.7 The do-while Loop

Sunday, February 5, 2023 5:27 PM

- Variation of while loop, syntax:
- ```
  do
  {
    // Loop body;
    Statement(s);
  } while (loop-continuation-condition);
  ```
- Loop body executed first, then loop-continuation-condition evaled, if true then loop body run again
- Main diff is order
  ```

5.8 The for Loop

Sunday, February 5, 2023 5:41 PM

- For loop has:
  - Initial-action = usually initializes a control variable
  - Loop-continuation-condition = test whether the ctrl var reached the ctrl var
  - Action-after-each-iteration = usually increments or decrements the ctrl var
  -
  - Then the body
-
-
```
for (int i = 0; i < 3; i++)
{
  cout << ("Welcome to C++!\n");
}
```
-
- The **action-after-each-iteration** in a **for** loop can be a list of zero or more comma-separated statements. For example:

```
for (int i = 1; i < 100; cout << i << endl, i++);
```

- This example is correct, but it is a bad example, because it makes the code difficult to read. Normally, you declare and initialize a control variable as an initial action and increment or decrement the control variable as an action after each iteration.
-
-

5.9 Which Loop to Use?

Sunday, February 5, 2023 5:58 PM

- While loop and for loop called pretest loops
- Do-while loop called posttest loop

5.10 Nested Loop

Sunday, February 5, 2023 6:25 PM

- Outer loop and inner loops
- Each time outer loop iterated, inner loop starts again
- Yep
- Inner loop go brr until dun, then outer lop fin its stuff, then if run again, outer loop run again, then inner loop go brr just like b4
- Wow
- Can take long time to run

5.11 Minimizing Numeric Errors

Sunday, February 5, 2023 6:42 PM

- Floating pt numbs are repped by approximation
- Use integer count to make sure all numbs added
-

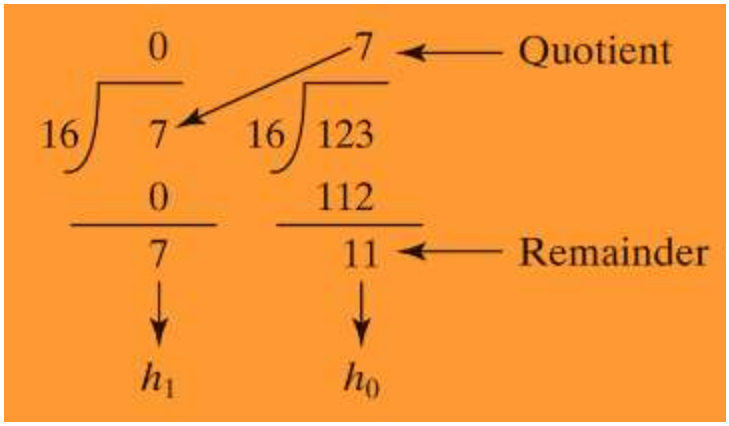
5.12 Case Studies

Sunday, February 5, 2023 6:59 PM

- Finding GCD
  - ```
    int gcd = 1; // Initial gcd is 1
    int k = 2; // Possible gcd

    while (k <= n1 && k <= n2)
    {
      if (n1 % k == 0 && n2 % k == 0)
        gcd = k; // Update gcd
      k++; // Next possible gcd
    }

    // After the loop, gcd is the greatest common divisor for n1 and n2
    ```
  - ```
    for (int k = 2; k <= n1 / 2 && k <= n2 / 2; k++)
    {
      if (n1 % k == 0 && n2 % k == 0)
        gcd = k;
    }
    ```
- Predict Future Tuition
  - ```
    double tuition = 10000;   int year = 0; // Year 0
    tuition = tuition * 1.07; year++;       // Year 1
    tuition = tuition * 1.07; year++;       // Year 2
    tuition = tuition * 1.07; year++;       // Year 3
    ```
  - ```
    double tuition = 10000; // Year 0
    int year = 0;
    while (tuition < 20000)
    {
      tuition = tuition * 1.07;
      year++;
    }
    ```
- Converting Decimals to Hex
  - $d = h_n \times 16^n + h_{n-1} \times 16^{n-1} + h_{n-2} \times 16^{n-2} + \ldots + h_2 \times 16^2 + h_1 \times 16^1 + h_0 \times 16^0$

- 
0
7
Quotient
16
7
16
123
0
112
7
11
Remainder
h_1
h_0
-

5.13 Keywords break & continue

Sunday, February 5, 2023 7:03 PM

- Both break and continue can be used in loop statements to give additional ctrls, don't overuse them tho
- Can use break in loop to immediately stop the loop
- Use continue keywork, ends current iteration and goes to end of the loop body
- So break breaks out of loop, continue breaks out of iteration
- Continue statement is always inside a loop
- Only use them if they simply the code and make it easier to read
-

```
int factor = 2;
while (factor <= n)
{
  if (n % factor == 0)
    break;
  factor++;
}
cout << "The smallest factor other than 1 for "
  << n << " is " << factor << endl;
```

-

5.14 Case Study: Checking Palindromes

Sunday, February 5, 2023 7:20 PM

- Same forward and back
- Can check by doing first and last check, see if they same, then move inward
-

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
  // Prompt user input
  cout << "Enter a string: ";
  string s;
  getline(cin, s);

  // low index
  int low = 0;

  // high index
  int high = s.length() - 1;

  bool isPalindrome = true;
  while (low < high)
  {
    if (s[low] != s[high])
    {
      isPalindrome = false; // Not a palindrome
      break;
    }

    low++;
    high--;
  }

  if (isPalindrome)
    cout << s << " is a palindrome" << endl;
  else
    cout << s << " is not a palindrome" << endl;

  return 0;
}
```

-

5.15 Case Study: Displaying Prime Numbers

Sunday, February 5, 2023 7:24 PM

- Pgrm displays first 50 prime numbers in 5 lines
- Prime if only divisor is 1 and itself
- Break pgrm apart:
  - Finde whether given numb is prime
  - Test whether it is prime
  - Count the prime numbers
  - Display each prime number, display 10 numb per line
- Need loop to test whether new numb is prime, if it is, increase count by 1, initially 0, when reaches 50, it stops

```
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
  const int NUMBER_OF_PRIMES = 50; // Number of primes to display
  const int NUMBER_OF_PRIMES_PER_LINE = 10; // Display 10 per line
  int count = 0; // Count the number of prime numbers
  int number = 2; // A number to be tested for primeness

  cout << "The first " << NUMBER_OF_PRIMES << " prime numbers are \n";

  // Repeatedly find prime numbers
  while (count < NUMBER_OF_PRIMES)
  {
    // Assume the number is prime
    bool isPrime = true; // Is the current number prime?

    // Test if number is prime
    for (int divisor = 2; divisor <= number / 2; divisor++)
    {
      if (number % divisor == 0)
      {
        // If true, the number is not prime
        isPrime = false; // Set isPrime to false
        break; // Exit the for loop
      }
    }

    // Display the prime number and increase the count
    if (isPrime)
    {
      count++; // Increase the count

       if (count % NUMBER_OF_PRIMES_PER_LINE == 0)
         // Display the number and advance to the new line
         cout << setw(4) << number << endl;
       else
         cout << setw(4) << number;
     }

    // Check if the next number is prime
    number++;
   }

   return 0;
}
```

- Check if prime by divisible btwn numbs 2 and numb/2 inclusive

Chapter Summary

Sunday, February 5, 2023 7:39 PM

1. There are three types of repetition statements: the while loop, the do-while loop, and the for loop.
2. The part of the loop that contains the statements to be repeated is called the loop body.
3. A one-time execution of a loop body is referred to as an iteration of the loop.
4. An infinite loop is a loop statement that executes infinitely.
5. In designing loops, you need to consider both the loop control structure and the loop body.
6. The while loop checks the loop-continuation-condition first. If the condition is true, the loop body is executed; if it is false, the loop terminates.
7. The do-while loop is similar to the while loop, except that the do-while loop executes the loop body first and then checks the loop-continuation-condition to decide whether to continue or to terminate.
8. The while loop and the do-while loop often are used when the number of repetitions is not predetermined.
9. A sentinel value is a special value that signifies the end of the loop.
10. The for loop generally is used to execute a loop body a fixed number of times.
11. The for loop control has three parts. The first part is an initial action that often initializes a control variable. The second part, the loop-continuation-condition, determines whether the loop body is to be executed. The third part is executed after each iteration and is often used to adjust the control variable. Usually, the loop control variables are initialized and changed in the control structure.
12. The while loop and for loop are called pretest loops because the continuation condition is checked before the loop body is executed.
13. The do-while loop is called a posttest loop because the condition is checked after the loop body is executed.
14. Two keywords, break and continue, can be used in a loop.
15. The break keyword immediately ends the innermost loop, which contains the break.
16. The continue keyword only ends the current iteration.