

Break statements

Prof. Hiren Patel, Ph.D.
Douglas Wilhelm Harder, M.Math.
hdpatel@uwaterloo.ca, dwharder@uwaterloo.ca
© 2015 by Douglas Wilhelm Harder and Hiren Patel.
Some rights reserved.

Outline

- In this lesson, we will:
 - Revisit the three looping statements
 - Describe the break statement
 - Look at the implementation
 - Consider some consequences

Looping statements

- We have seen three looping statements
 - While statements
 - For statements
 - Do-while statements
- It is only possible to end the looping when the associated condition is checked and evaluates to `false`
 - This can sometimes lead to interesting code

Looping statements

- Suppose we have a while loop

```

while ( condition-1 ) {
    // Do something...
}

if ( condition-2 ) {
    // Do something else...
} else {
    // Do something yet again...
    // Determine it is time to end this loop...
}

// Continue doing "stuff"...
}

```

Break statements

- A break statement:


```
break;
```

 ends the inner-most looping statement that contains it
 - It takes no arguments, has no return value, etc.
 - It is always a statement by itself

A goto statement in disguise?

- Some consider a break statement to be a goto statement


```
while ( condition-1 ) {
    // Do something...

    if ( condition-2 ) {
        // Do something else...
    } else {
        // Do something yet again...
        // Determine it is time to end this loop...
        if ( condition-3 ) {
            goto end_of_loop;
        }
    }
}

// Continue doing "stuff"...
```

```
// Next statement following the while statement
end_of_loop: next-statement;
```

Break statements

- For example:


```
while ( condition-1 ) {
    // Do something...

    if ( condition-2 ) {
        // Do something else...
    } else {
        // Do something yet again...
        // Determine it is time to end this loop...
        if ( condition-3 ) {
            break;
        }
    }
}

// Continue doing "stuff"...

// Next statement following the while statement
```

A goto statement in disguise?

- When you document a looping statement, you must also document any break statements found within it
- Some organize ban the use of a break statement in a similar way that they ban goto statements
 - Excessive or irresponsible use of a break statement can lead to programmer confusion

Example

- If you are repeatedly seeking input, this may be reasonable:

```
int main() {
    // This loop terminates as soon as the user enters '0'
    while ( true ) {
        // Do some setup...

        std::cout << "Enter a number ('0' to quit): ";
        std::cin >> n;

        if ( n == 0 ) {
            break;
        }

        // Do something with 'n'
    }

    return 0;
}
```



One loop only

- The break statement only exits the inner-most loop

```
while ( condition-1 ) {
    // Do something

    while ( condition-2 ) {
        // Do something else...

        if ( condition-3 ) {
            // Do some final stuff...
            break;
        }

        // Continue doing stuff...
    }

    // Continue doing more stuff...
}

// Further statements
```

The break statement jumps here



One loop only

- To break out of nested looping statements, we must use goto

```
while ( condition-1 ) {
    // Do something

    while ( condition-2 ) {
        // Do something else...

        if ( condition-3 ) {
            // Do some final stuff...
            goto end_of_outer_loop;
        }

        // Continue doing stuff...
    }

    // Continue doing more stuff...
}

// Further statements
```

The goto statement jumps here
- appending it prevents other programmers
from inserting code between the end of
the loop and the next statement



Summary

- Following this lesson, you now
 - Understand the break statement
 - Know how it interacts with looping statements
 - It only exits the inner-most looping statement
 - See that it is a simplified goto statement



References

[1] No references?



Colophon

These slides were prepared using the Georgia typeface. Mathematical equations use Times New Roman, and source code is presented using Consolas.

The photographs of lilacs in bloom appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens on May 27, 2018 by Douglas Wilhelm Harder. Please see

<https://www.rbg.ca/>

for more information.



Disclaimer

These slides are provided for the ECE 150 *Fundamentals of Programming* course taught at the University of Waterloo. The material in it reflects the authors' best judgment in light of the information available to them at the time of preparation. Any reliance on these course slides by any party for any other purpose are the responsibility of such parties. The authors accept no responsibility for damages, if any, suffered by any party as a result of decisions made or actions based on these course slides for any other purpose than that for which it was intended.

