

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
Department of Electrical & Computer Engineering

ECE 150 *Fundamentals of Programming*

# Using member functions

Douglas Wilhelm Harder, M.Math. LEL.  
Prof. Hiren Patel, Ph.D., P.Eng.  
Prof. Werner Diel, Ph.D.

© 2018 by Douglas Wilhelm Harder and Hiren Patel. Some rights reserved.

CC BY NC SA

1

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
Department of Electrical & Computer Engineering

Using member functions

## Outline

- In this lesson, we will:
  - See how to use member functions
  - Look at using member functions for the
    - `std::cout` object
    - Instances of the `std::string` class
    - Introduce the `std::set` class and its member functions
  - Observe that we don't have to know about the constructors and destructors

CC BY NC SA

2

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
Department of Electrical & Computer Engineering

Using member functions

## Member functions

- We discussed how we don't want users accessing member variables
  - Instead, users should only be able to call functions associated with the class
  - Inside a function, you can make sure everything is correct when the function returns
- How do we allow the author of the class to write functions that access member variables, but at the same time disallow users from writing similar functions?
  - Like member variables, a class can also have *member functions*
- With member functions, we will call them in the exact way that we access member variables:
  - with the `.` operator, only now we pass it arguments, as well

CC BY NC SA

3

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
Department of Electrical & Computer Engineering

Using member functions

## Calling member functions

- Here is how member functions are called:

```
int main() {
    std::cout << M_PI << std::endl;
    std::cout.precision( 16 );
    std::cout << M_PI << std::endl;
    std::cout << std::sin( M_PI ) << std::endl;

    std::cout.width( 10 );
    std::cout << 42 << std::endl;
    std::cout << 42 << std::endl;

    std::cout.fill( '?' );
    std::cout.width( 10 );
    std::cout << 42 << std::endl;
    std::cout << 42 << std::endl;

    return 0;
}
```

Output:

```
3.14159
3.141592653589793
1.224646799147353e-16
42
42
???????42
42
```

CC BY NC SA

4

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
Department of Electrical and Computer Engineering

Using member functions 5

## Calling member functions

- Here is how member functions are called:

```
int main() {
    std::string greeting( "G'day, for those of you..." );
    std::cout << greeting << std::endl;

    greeting.resize( greeting.length() - 3 ); // Reduce length by three
    std::cout << greeting << std::endl;

    greeting += " who don't know us...";
    std::cout << greeting << std::endl;

    greeting.clear();
    std::cout << "\"" << greeting << "\"" << std::endl;

    return 0;
}
```

**Output:**  
G'day, for those of you...  
G'day, for those of you  
G'day, for those of you who don't know us...  
""



5

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
Department of Electrical and Computer Engineering

Using member functions 6

## Calling member functions

- A call to a member function is a call on that specific object:

```
int main() {
    std::string line1{ "Goose, goose, goose," };
    std::string line2{ "You bend your neck towards the sky and sing." };
    std::string line3{ "Your white feathers float on the green water," };
    std::string line4{ "Your red feet push the clear waves." };

    std::cout << line1.length() << std::endl;
    std::cout << line2.length() << std::endl;
    std::cout << line3.length() << std::endl;
    std::cout << line4.length() << std::endl;

    line3.replace( 33, 5, "emerald" );

    std::cout << line3.length() << std::endl;
    std::cout << line3 << std::endl;

    return 0;
}
```

**Output:**  
20  
44  
45  
35  
Your ... the emerald water,  
47



6

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
Department of Electrical and Computer Engineering

Using member functions 7

## Understanding member functions

- Given a class, once you know what the member functions are supposed to perform, you can now use the class without knowing anything about how that class is implemented

```
#include <set>
#include <iostream>

int main() {
    std::set<int> data;

    for ( unsigned int k{0}; k < 10; ++k ) {
        int n{};
        std::cin >> n;
        data.insert( n );
    }

    std::cout << "Number of entries: " << data.size() << std::endl;
    std::cout << "Instances of 65: " << data.count( 65 ) << std::endl;
}
```

**Input:**  
34 15 65 49 91 42 83 19 70 65

**Output:**  
Number of entries: 9  
Instances of 65: 1



7

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
Department of Electrical and Computer Engineering

Using member functions 8

## Understanding member functions

```
std::cout << data.erase( 65 ) << std::endl;
std::cout << "Number of entries: " << data.size() << std::endl;
std::cout << data.erase( 65 ) << std::endl;
std::cout << "Number of entries: " << data.size() << std::endl;

for ( std::set<int>::iterator itr{ data.begin() };
      itr != data.end(); ++itr ) {
    std::cout << " " << *itr;
}

return 0;
}
```

**Output:**  
1  
Number of entries: 8  
0  
Number of entries: 8  
15 19 34 42 49 70 83 91



8

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
FACULTY OF INFORMATION TECHNOLOGY

Using member functions 9

## The standard library

- The site cplusplus.com has an excellent interface for documenting the various classes and corresponding member functions of the standard library:



9

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
FACULTY OF INFORMATION TECHNOLOGY

Using member functions 10

## Constructors and destructors?

- We emphasized there are three aspects that we must deal with:
  - The creation and initialization Constructors
  - The lifetime of the object Member functions
  - The clean-up prior to destruction Destructors
- We didn't have to deal with the constructors or the destructors because the compiler deal with these:
  - The constructor was called when the local variable was declared
  - The destructor was called when the local variable went out of scope



10

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
FACULTY OF INFORMATION TECHNOLOGY

Using member functions 11

## Summary

- Following this lesson, you now
  - Understand how to use member functions
  - Have seen member functions for:
    - `std::cout`
    - `std::string`
    - `std::set`
  - Understand that the compiler schedules calls to the constructor and the destructor when a local variable is declared and when it goes out of scope



11

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
FACULTY OF INFORMATION TECHNOLOGY

Using member functions 12

## References

- [https://en.wikipedia.org/wiki/C++\\_classes](https://en.wikipedia.org/wiki/C++_classes)
- [https://en.wikipedia.org/wiki/Standard\\_Template\\_Library](https://en.wikipedia.org/wiki/Standard_Template_Library)



12



## 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.



13



## 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.



14

