

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

ECE 150 *Fundamentals of Programming*

A windows class for graphical user interfaces

CC BY NC SA

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

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

1

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

A windows class for graphical user interfaces

Outline

- In this lesson, we will:
 - Describe another application of inheritance
 - See the need for a common class that contains shared information
 - Describe subsequent benefits of such an approach
 - Describe how to declare a class as abstract

CC BY NC SA

2

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

A windows class for graphical user interfaces

Warning

- Please remember, we don't expect you know how to implement a graphical user interface
 - This is the result of years of experience and a team of software engineers and developers working together
- We are presenting this as an example of where inheritance could be useful in the real world
 - Try to understand why we are suggesting and describing

CC BY NC SA

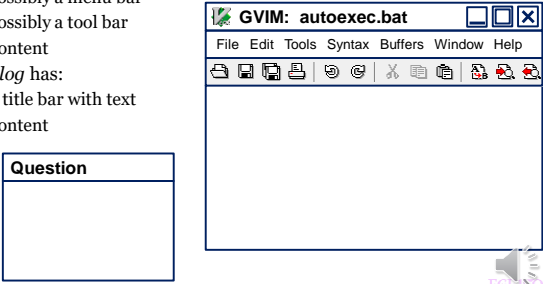
3

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

A windows class for graphical user interfaces

Graphical user interface

- A graphical user interface *window* can either be a *frame* or a *dialog*
 - A *frame* has:
 - A title bar with an icon, text, and color
 - Possibly a menu bar
 - Possibly a tool bar
 - Content
 - A *dialog* has:
 - A title bar with text
 - Content



CC BY NC SA

4

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Faculty of Information Arts
Winter 2019

A windows class for graphical user interfaces 5

Graphical user interface

- Each of these has information that must be stored in order to correctly display it
 - Thus, we would have two separate classes:


```
class Frame;
class Dialog;
```



5

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Faculty of Information Arts
Winter 2019

A windows class for graphical user interfaces 6

A comment on constructors

- A constructor takes a number of arguments, but too many will overwhelm the user leading to errors:

```
Frame main_frame{ "Hello world",
  Font{ "Times New Roman", 16 },
  "my_icon.png",
  100, 70, 593, 231,
  RGB_Color{ 15, 25, 193 },
  RGB_Color{ 255, 255, 255 },
  true, false, false, false, true, 10, 10 };
```

- Consequently, many parameters automatically have default values that can only be set via member functions



6

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Faculty of Information Arts
Winter 2019

A windows class for graphical user interfaces 7

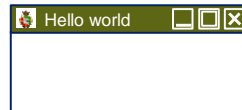
An example of displaying a frame

- Here is a possible minimal application using a frame:

```
int main() {
  Frame main_frame{};
  main_frame.title_text( "Hello world!" );
  main_frame.title_background_color( RGB_color{ 93, 102, 21 } );
  main_frame.title_text_color( RGB_color{ 255, 255, 255 } );
  main_frame.title_icon( "image.png" );
  // No menu bar unless one is added
  // No tool bar unless one is added

  main_frame.display();

  return 0;
}
```



7

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Faculty of Information Arts
Winter 2019

A windows class for graphical user interfaces 8

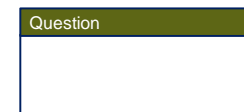
An example of displaying a dialog

- Here is a minimal implementation of a dialog:

```
int main() {
  Dialog query{};
  query.title_text( "Question" );
  query.title_background_color( RGB_color{ 93, 102, 21 } );
  query.title_text_color( RGB_color{ 255, 255, 255 } );

  Return_type value{ query.display() };

  return 0;
}
```



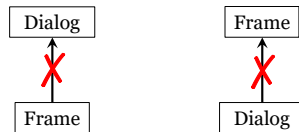
8

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
FACULTY OF INFORMATION AND COMMUNICATIONS

A windows class for graphical user interfaces 9

Is one derived from the other?

- There are significant differences between frames and dialogs
 - Dialogs are modal and have return values
 - Frames have menu bars and tool bars



- However, there are some features that are in common
 - The title bar background color, text and text color
 - The member functions used to set these
 - The display member function



9

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
FACULTY OF INFORMATION AND COMMUNICATIONS

A windows class for graphical user interfaces 10

Common features

```
class Frame {
public:
    Return_type display();
    // Other common member functions
    // Other member functions
private:
    Image    title_bar_icon_;
    RGB_color title_bar_background_color_;
    std::string title_bar_text_;
    RGB_color title_bar_text_color_;
    Menu_bar menu_bar_;
    Tool_bar tool_bar_;
    Panel    content_;
};

class Dialog {
public:
    Return_type display();
    // Other common member functions
    // Other member functions
private:
    RGB_color title_bar_background_color_;
    std::string title_bar_text_;
    RGB_color title_bar_text_color_;
    Panel    content_;
};
```



10

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
FACULTY OF INFORMATION AND COMMUNICATIONS

A windows class for graphical user interfaces 11

Common features

- Compare these two classes:

```
class Window {
public:
private:
};
```

```
class Frame { public Window {
public:
    Return_type display();
    // Other common member functions
    // Other member functions
private:
    Image    title_bar_icon_;
    RGB_color title_bar_background_color_;
    std::string title_bar_text_;
    Menu_bar menu_bar_;
    Tool_bar tool_bar_;
    Panel    content_;
};
```

```
class Dialog { public Window {
public:
    Return_type display();
    // Other common member functions
    // Other member functions
private:
    RGB_color title_bar_background_color_;
    std::string title_bar_text_;
    Panel    content_;
};
```



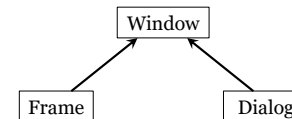
11

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
FACULTY OF INFORMATION AND COMMUNICATIONS

A windows class for graphical user interfaces 12

Common features

- Thus, our class diagram would be as follow:



- In a next topic, we will describe how we can use inheritance to display the content of both frames and dialogs



12

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Faculty of Information Systems
13

A windows class for graphical user interfaces

Benefit of inheritance

- One great benefit of inheritance is uniformity:
 - Suppose with your next release, you include additional features:
 - A title text typeface
 - Helvetica, Georgia, Cambria, etc.
 - A title text style
 - Italicized
 - A title text weight
 - Bold
 - A title text size
 - Without a common base class, you would implement all these features in each class separately
 - You may forget, or may not have sufficient time to do both
 - This will annoy users
 - Doing it once in the base class includes these changes in both derived classes



13

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Faculty of Information Systems
14

A windows class for graphical user interfaces

Can we create a window?

- Can we declare an instance of a window?
 - There is no point in declaring an instance of a window class, as the window class cannot be displayed
 - We can assign the `display()` function the value `0`
 - This declares this class as being an *abstract class*
 - That is, it is an idea, and not something that can be concretely implemented
 - Derived classes must implement their versions of this member function to allow users to declare instances of these classes

```
class Window {
public:
    Return_type display() = 0;
    // Other common member functions
private:
    // Common member variables
};
```



14

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Faculty of Information Systems
15

A windows class for graphical user interfaces

Summary

- Following this lesson, you now
 - Have seen another application of inheritance
 - Understand the benefit of creating a common base class
 - Discussed abstract classes that cannot be declared in a program



15

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Faculty of Information Systems
16

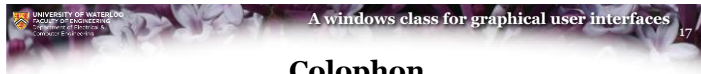
A windows class for graphical user interfaces

References

- [1] https://en.wikipedia.org/wiki/Linked_list
- [2] [https://en.wikipedia.org/wiki/Node_\(computer_science\)](https://en.wikipedia.org/wiki/Node_(computer_science))



16



Colophon

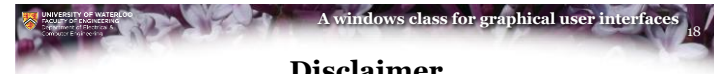
These slides were prepared using the Georgia typeface. Mathematical equations use Times New Roman, and source code is presented using Consolas.

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.



17



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.



18

