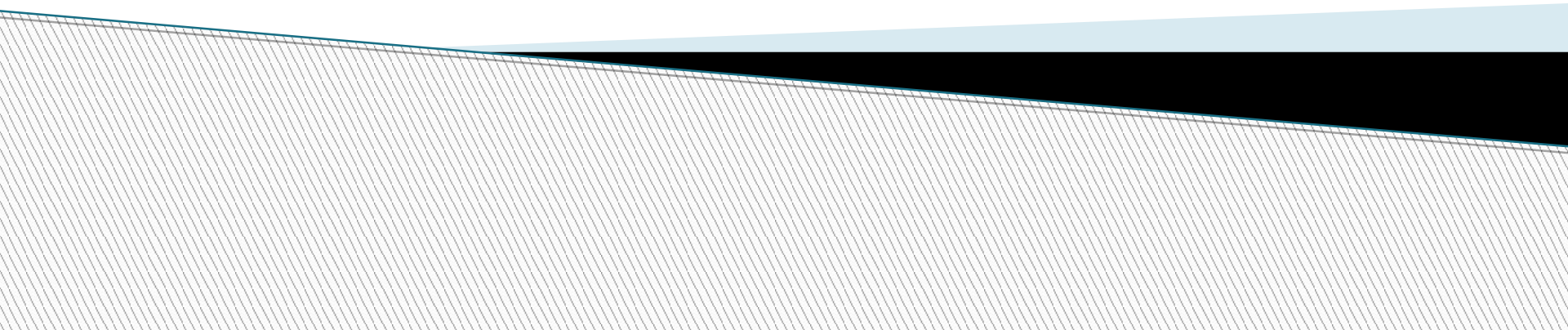


# **41012 Programming for Mechatronic Systems**

Week 2




# Overview

- Outstanding questions from last week
- We continue looking at pointers and functions
- Push into OO methodology with
  - ▢ Classes – Objects - Constructors
  - ▢ Scope of variables
  - ▢ Access specifiers

# but first... the terminal

- ▶ Text based interface to the operating system
  - Faster and more powerful than GUI
  - A bit trickier to learn
- ▶ Common commands:
  - ls – list files
  - cd – change directory
  - cp – copy file
  - mv – move/rename file
  - pwd - where am I in the filesystem?
  - lots more...
- ▶ Other tips:
  - Ctrl+Alt+t – open a terminal window
  - Ctrl+Shift+c – copy
  - Ctrl+Shift+v – paste
  - Middle click – copy and paste current selection
  - man <command> - get help!?!

# Header / Source File

- ▶ Why needed?
    - It allows you to separate *interface* from *implementation*
    - It speeds up compile time
  - ▶ Generally the interface (header) file is all that is needed to use an external library
  - ▶ Allows library implementation to be hidden
  - ▶ Further Details
  - ▶ <http://www.cplusplus.com/articles/Gw6AC542/>
- 

# Classes

## ▶ Task:

- Implement the following class
- Create an executable uses the class to compute the area and perimeter of a rectangle

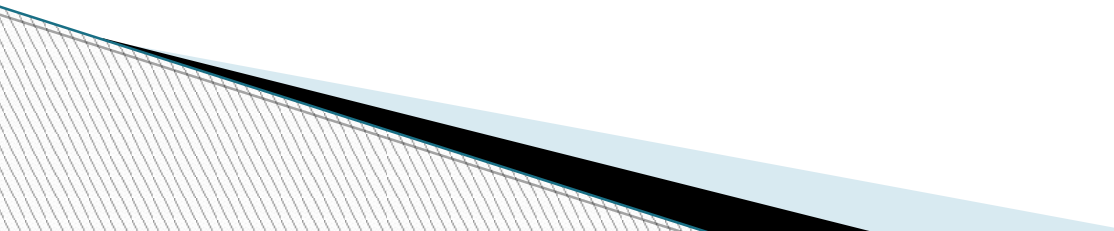
```
class Rectangle {  
public:  
    void setWidthAndHeight(int,int);  
    int area(void);  
    int perimeter(void);  
private:  
    int width_, height_;  
};
```

## Questions:

- ▶ What are the functions of this class?
- ▶ What do we call functions of a class?
- ▶ What arguments do they accept?
- ▶ What are their return types?
- ▶ What are the access specifiers?
- ▶ Why do we use them?
- ▶ Why do we have the SetWidthAndHeight() method?
- ▶ Can you see any problems with this class?
- ▶ Where should we place the implementation of the class?
- ▶ What do we need to alter in CMakeLists.txt to compile the executable?

# Classes

```
class Rectangle {  
    Rectangle();  
    ~Rectangle();  
public:  
    void setWidthHeight(double width, double height);  
    double area(void);  
    double perimeter(void);  
private:  
    double width_, height_;  
};
```



# Classes

```
Rectangle::Rectangle():width_(0.0),height_(0.0)
{
}

void Rectangle::setWidthHeight(double width, double height)
{
    width_ = width;
    height_ = height;
}

double Rectangle::area()
{
    return width_ * height_;
}

double Rectangle::perimeter()
{
    return 2*width_ + 2*height_;
}
```

Questions:

- ▶ What problem does this modification solve?
- ▶ Any other problems?

# Classes

- ▶ Create a class that:
  - In Constructor
    - a) Accepts a Seed for random Number Generator
    - b) N - number specifying how many random numbers are generated each time
  - Has a Function
    - a) Accepts an array and populates the array with N numbers generated on each time
    - b) Has a function that accepts N

## Questions:

- ▶ What are the pinch points here, how to ensure the program does not cause a segmentation fault?
  - ▶ What is a segmentation fault?
  - ▶ Why do we have N in constructor and a function
  - ▶ Does the main of the class ultimately define the max length of the array
  - ▶ Is there any way to make this allocation dynamic
- 