Lecture 3: String Theory

Bart Iver van Blokland

(Rune Sætre)

Previously on…

Season 48

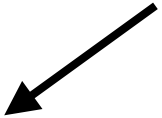
What On Earth Was That Lecturer Talking About

Last lecture

Basics of the C++ Language

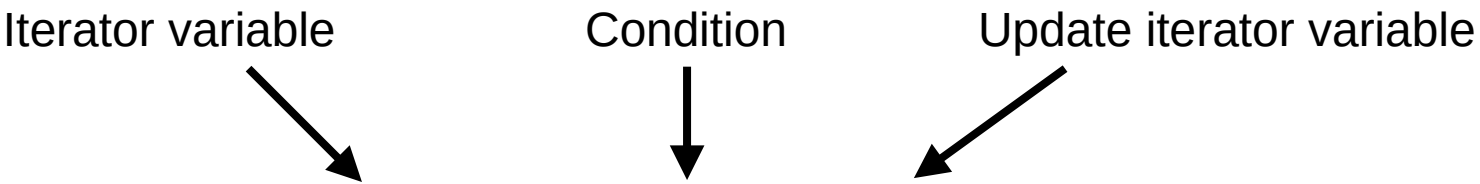
- Variables
- Data types
- Operators
- Text input
- Type casting
- If statements
- Loops
- Functions

Due this Friday!!



(all you need to complete assignment 1)

For loops

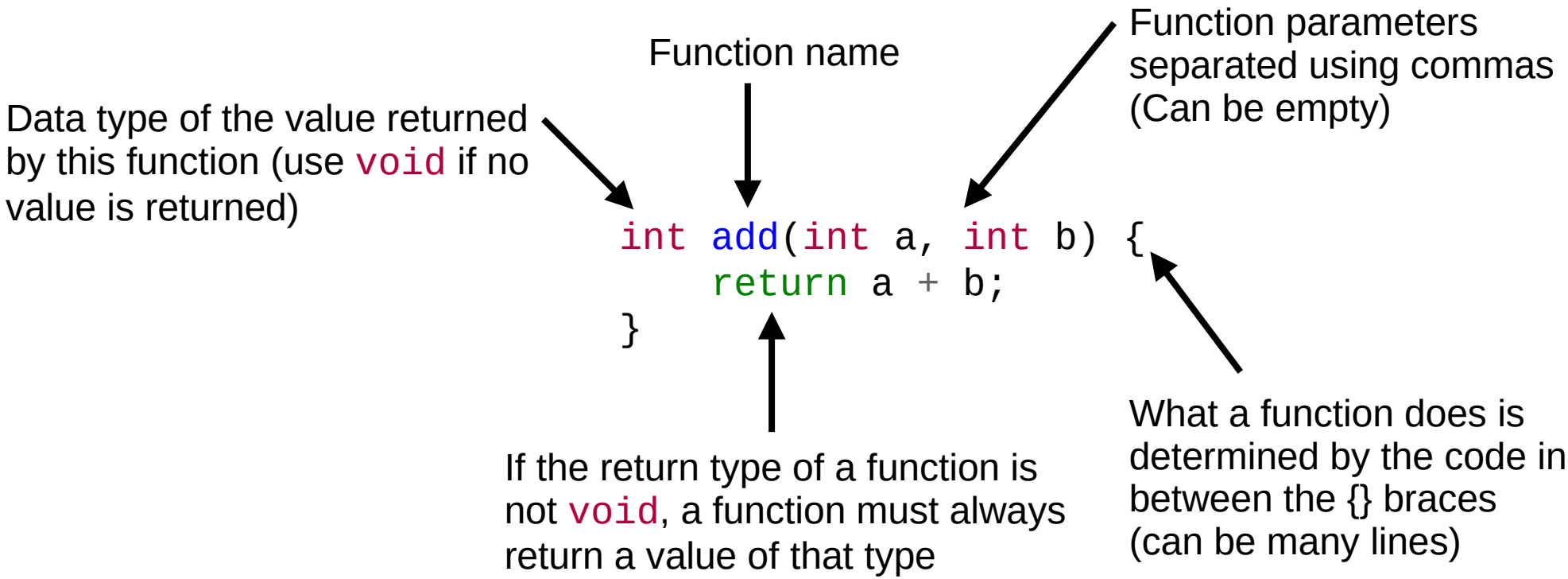



```
for(int i = 0; i < 10; i++) {
    // in this example, these lines
    // are repeated 10 times
}
```

How a for loop works:

1) Create iterator variable
2) Evaluate condition
   - If false: continue the program after the loop
   - If true: run the code in between the {} brackets
3) Update iterator variable
4) Go back to step 2

Functions


Data type of the value returned by this function (use void if no value is returned)
Function name
Function parameters separated using commas (Can be empty)
int add(int a, int b) {
return a + b;
}
If the return type of a function is not void, a function must always return a value of that type
What a function does is determined by the code in between the {} braces (can be many lines)

Today

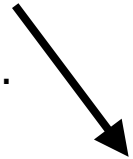
We will start on assignment 2!

- **More on functions**
- Strings
- Basics of graphics
- Some tips for writing code

Functions: Default parameters

It is possible to give function parameters a default value.

If the parameter is omitted, the default value is used.



```
int increment(int value, int incrementBy = 1) {
    return value + incrementBy;
}

int main() {
    cout << increment(10) << endl;
    cout << increment(10, 5) << endl;
    return 0;
}
```

Prints 11

Prints 15

Functions: Function overloading

Function overloading: defining multiple functions with the same name.

The compiler determines based on the data type of the parameters which function to call.

```
int add(int a, int b) {
    return a + b;
}

double add(double a, double b) {
    return a + b;
}

int main() {
    cout << "5 + 6 = " << add(5, 6) << endl;
    cout << "5.0 + 6.0 = " << add(5.0, 6.0) << endl;
    cout << "5.0 + 6.0 = " << add(5.0, 6) << endl;
    return 0;
}
```

Two integers, calls the top variant

Two doubles, calls the bottom variant

Error: not sure which one to use

Today

We will start on assignment 2!

- More on functions
- **Strings**
- Basics of graphics
- Some tips for writing code

Converting to and from string

Category	Data Type	Convert from string	Convert to string
Integers	`char`	`stoi(value)`	`to_string(int(value))`
	`unsigned char`	`stoul(value)`	`to_string(int(value))`
	`short`	`stoi(value)`	`to_string(value)`
	`unsigned short`	`stoul(value)`	`to_string(value)`
	`int`	`stoi(value)`	`to_string(value)`
	`unsigned int`	`stoul(value)`	`to_string(value)`
	`long long`	`stoll(value)`	`to_string(value)`
	`unsigned long long`	`stoull(value)`	`to_string(value)`
Real numbers	`float`	`stof(value)`	`to_string(value)`
	`double`	`stod(value)`	`to_string(value)`
Miscellaneous	`bool`	not applicable	`to_string(value)`
	`enum`	not applicable	`to_string(value)`

Examples:

```
string stringValue = "15";
int intValue = stoi(stringValue);
```

```
double score = 1337.3;
string converted = to_string(score);
```

Strings

```
string text;

text += 'H';
text += 'e';
text += 'y';

text += " there!";

text += "\n";


text += "Here is a number: "
        + to_string(42);

cout << text << endl;
```

Unlike numeric data types, strings are automatically initialised to an empty string.

You can append single characters to a string using single quotes (')

Or append another string

Want a fresh row? Use the special \n character to create a new line.

Use the `to_string()` function to convert numeric types to strings before appending them.

You can print a string using cout, just like any other basic type.

Windows and Norwegian characters

Windows

```
1   #include "std_lib_facilities.h"
2
3   int main() {
4       cout << "Går det an å bruke Norske tegn?" << endl;
5       return 0;
6   }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
G├Ñr det an ├Ñ bruke Norske tegn?
```

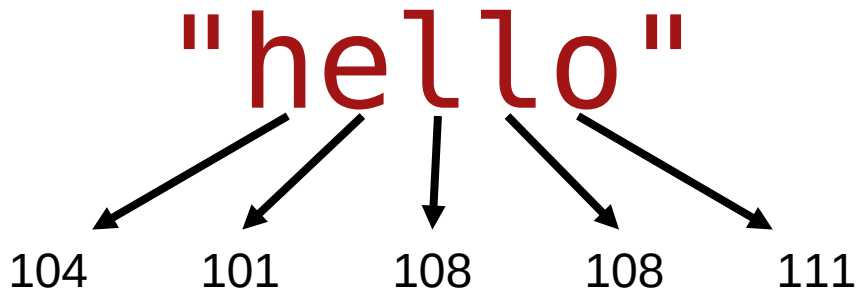
MacOS and Linux

```
1   #include "std_lib_facilities.h"
2
3   int main() {
4       cout << "Går det an å bruke Norske tegn?" << endl;
5       return 0;
6   }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Går det an å bruke Norske tegn?
```

Fundamental problem: computers can only store numbers



Solution: we assign a number to each character

How numbers are interpreted as characters determines how they are shown on screen.

Windows by default uses an encoding that does not support Norwegian characters.

Mac and Linux use Unicode (UTF-8), which does.

Today

We will start on assignment 2!

- More on functions
- Strings
- **Basics of graphics**
- Some tips for writing code

Graphics: AnimationWindow

- Used in most assignments
- Comes preinstalled with the VS Code extension
- A library we have written, not part of standard C++
- Detailed documentation:

  https://tdt4102.pages.stud.idi.ntnu.no/documentation/

Create a window

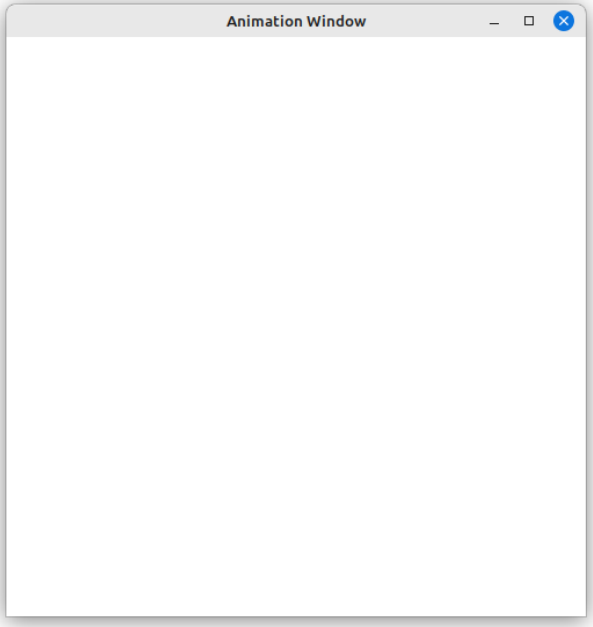
```
#include "std_lib_facilities.h"
#include "AnimationWindow.h"

int main() {
    AnimationWindow window;

    // Draw things here

    window.wait_for_close();
    return 0;
}
```

AnimationWindow must be imported before it can be used

Creating a variable of type AnimationWindow opens a window

We'll replace this with functions that draw shapes

Without using wait_for_close() the program will exit immediately and in the process close the window.

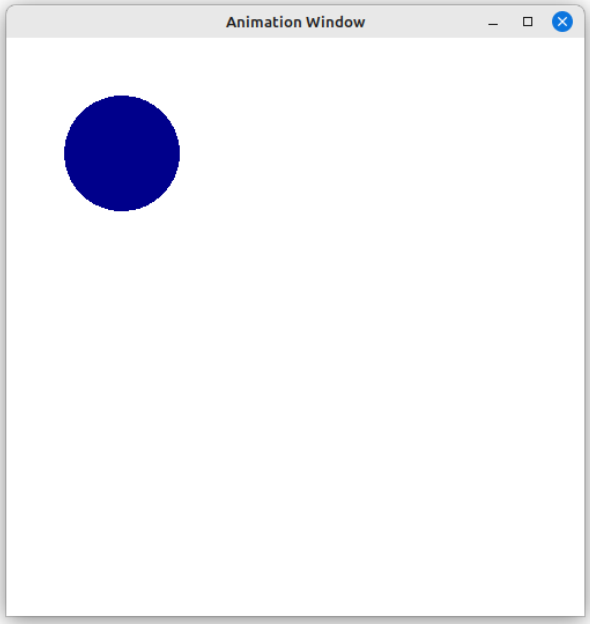


Drawing: Circles

```
#include "std_lib_facilities.h"
#include "AnimationWindow.h"

int main() {
    AnimationWindow window;

    window.draw_circle({100, 100}, 50);

    window.wait_for_close();
    return 0;
}
```

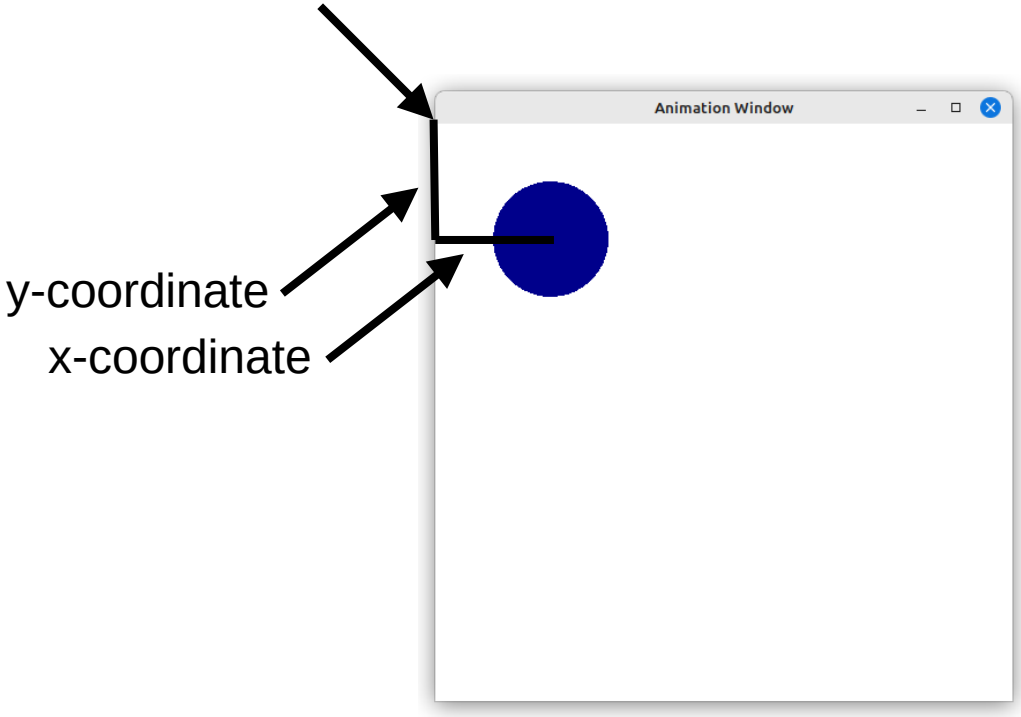
Radius of the circle

- Location where the circle should be drawn
- The function requires that we group the x and y coordinates using {}



Drawing: Coordinates

The origin (0, 0) is in the top left corner



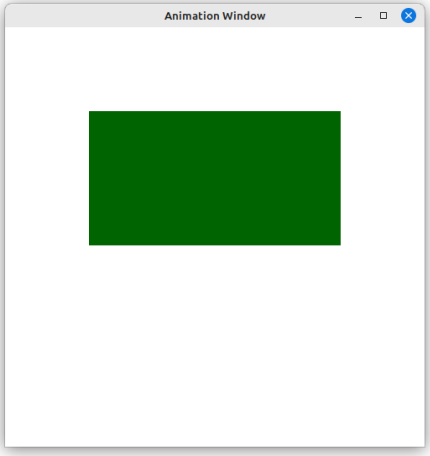
All coordinates are measured in pixels!

Drawing: Rectangles

```
#include "std_lib_facilities.h"
#include "AnimationWindow.h"

int main() {
    AnimationWindow window;

    window.draw_rectangle({100, 100}, 300, 160);

    window.wait_for_close();
    return 0;
}
```

Width Height

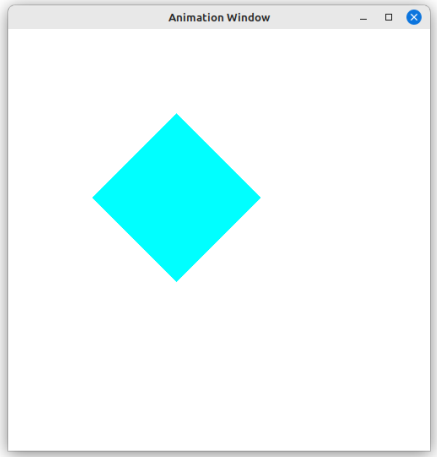
Location where the top left corner of the rectangle should be positioned



Drawing: Quads

```
#include "std_lib_facilities.h"
#include "AnimationWindow.h"

int main() {
    AnimationWindow window;

    window.draw_quad({200, 100}, {100, 200}, {200, 300}, {300, 200});

    window.wait_for_close();
    return 0;
}
```

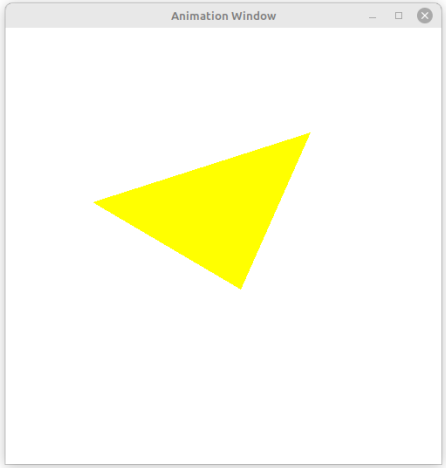
Vertex 1 Vertex 2 Vertex 3 Vertex 4



Drawing: Triangle

```
#include "std_lib_facilities.h"
#include "AnimationWindow.h"

int main() {
    AnimationWindow window;

    window.draw_triangle({350, 120}, {100, 200}, {270, 300});

    window.wait_for_close();
    return 0;
}
```

Vertex 1 Vertex 2 Vertex 3

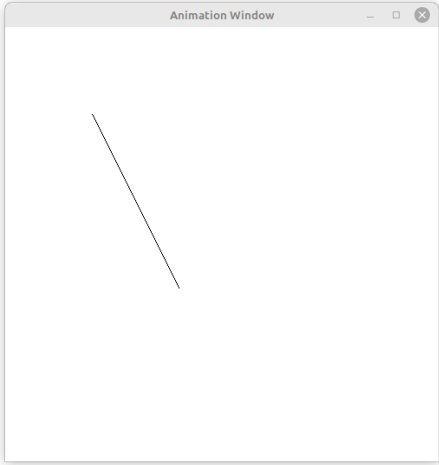


Drawing: Line

```
#include "std_lib_facilities.h"
#include "AnimationWindow.h"

int main() {
    AnimationWindow window;

    window.draw_line({100, 100}, {200, 300});

    window.wait_for_close();
    return 0;
}
```

Start point

End point



Drawing: Arc

```
#include "std_lib_facilities.h"
#include "AnimationWindow.h"

int main() {
    AnimationWindow window;

    window.draw_arc({250, 200}, 150, 150, 30, 200);

    window.wait_for_close();
    return 0;
}
```

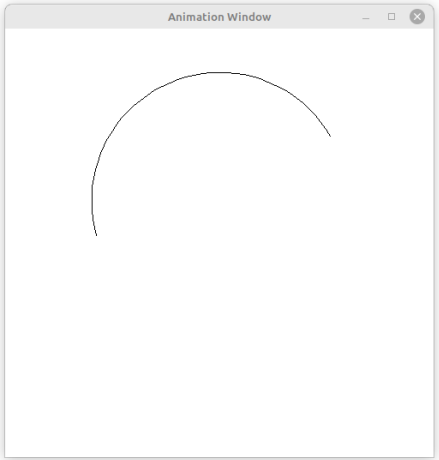
Centre point

Start angle

End angle

Width

Height

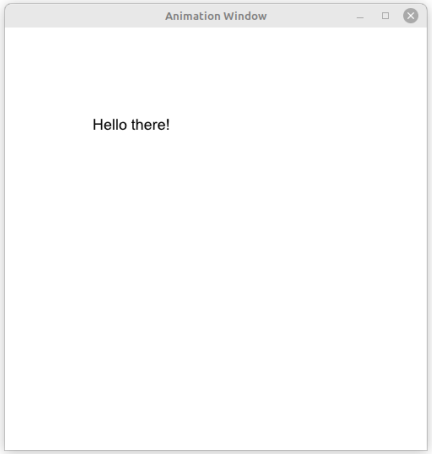


Drawing: Text

```
#include "std_lib_facilities.h"
#include "AnimationWindow.h"

int main() {
    AnimationWindow window;

    window.draw_text({100, 100}, "Hello there!");

    window.wait_for_close();
    return 0;
}
```

Starting point

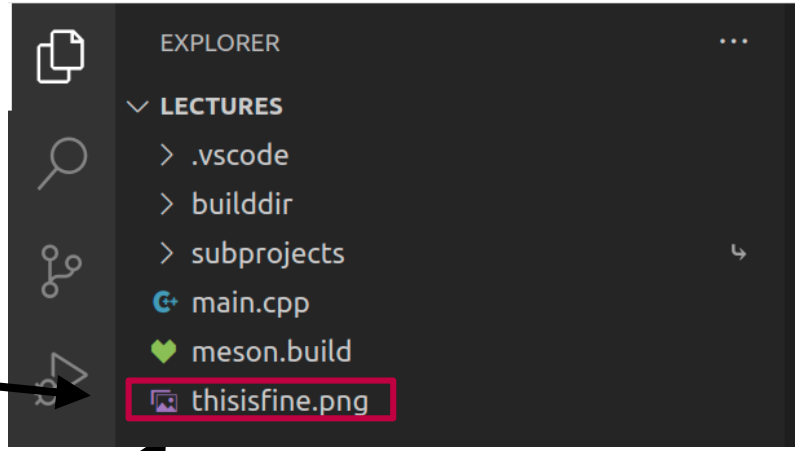
Text to show



Drawing: Images

```
#include "std_lib_facilities.h"
#include "AnimationWindow.h"

int main() {
    AnimationWindow window;

    Image image("thisisfine.png");
    window.draw_image({100, 100}, image);

    window.wait_for_close();
    return 0;
}
```

Name of the image file.
Must match exactly!



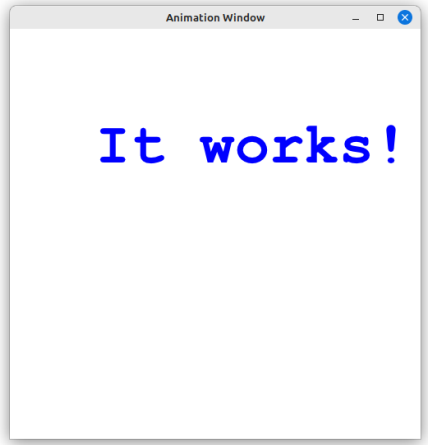
Where to draw the image

image to draw

Put images in your project directory

Drawing: optional parameters

```
#include "std_lib_facilities.h"
#include "AnimationWindow.h"

int main() {
    AnimationWindow window;


    window.draw_text({100, 100}, "It works!", Color::blue, 80,
                     Font::courier_bold);


    window.wait_for_close();
    return 0;
}
```



All shown drawing functions have optional parameters for changing things like colour. See the documentation for more information: https://tdt4102.pages.stud.idi.ntnu.no/documentation/

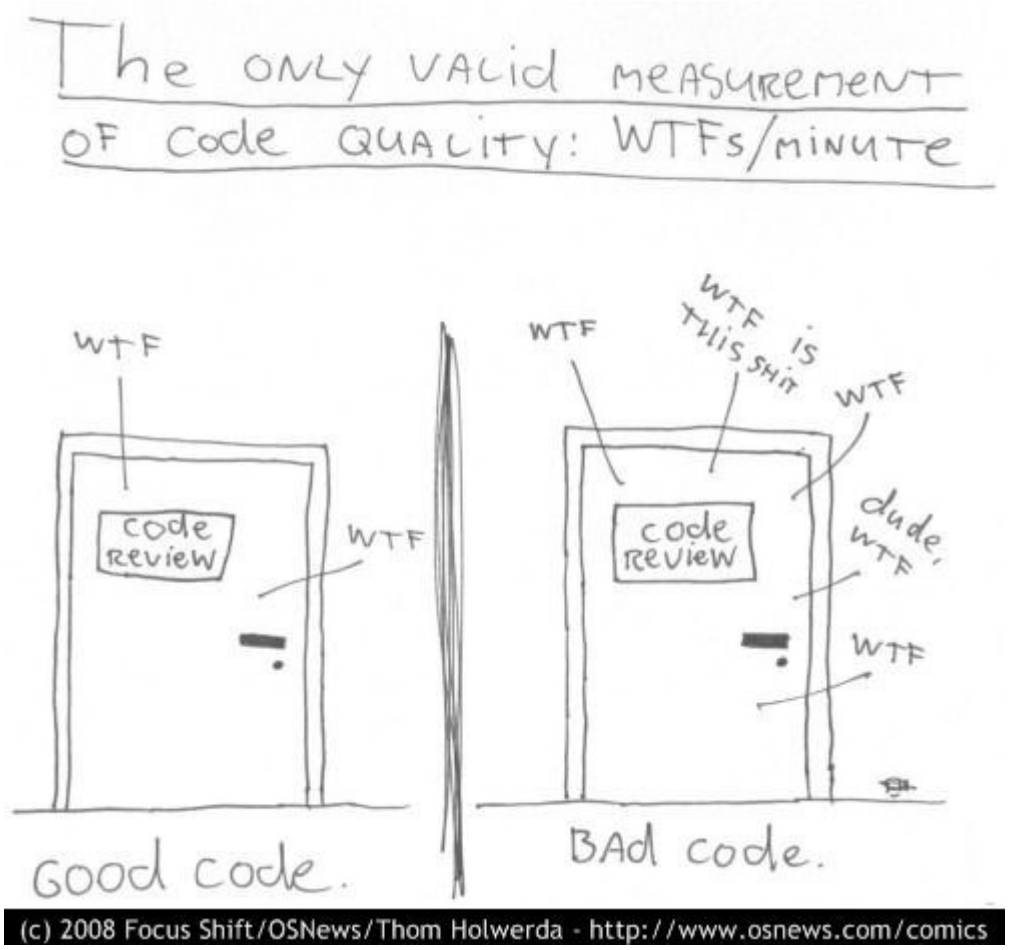
Today

We will start on assignment 2!

- Repetition and more on functions
- Strings
- Basics of graphics
- **Some tips for writing code**

Our code might work, but how well is it built?

Code is not always easy to understand



What does this function do?

```
int r(int x, int y) {
    int b = 0;
    for(int q = x; q < y; q++) {
        if(q % 2 == 1) {
            b++;
        }
    }
    return b;
}
```

What does this function do?

```
int countOddNumbersInRange(int start, int end) {
    int count = 0;
    for(int i = start; i < end; i++) {
        bool numberIsOdd = i % 2 == 1;
        if(numberIsOdd) {
            count++;
        }
    }
    return count;
}
```

Names help figure out what code does!

Names: Things you can do

- Communicate intent

```
int timeInDays;
int visitedPageCount;
bool buttonWasClicked;
```

- Names give meaning to values

```
if(distance > 42.195) {

}

double marathonDistanceKm = 42.195;
bool hasCompletedMarathon = distance > marathonDistanceKm;
if(hasCompletedMarathon) {

}
```

Names: Things you can do

- Don’t hide information

```
// hard to see the difference
double dailyTemperatureMeasureIncrementInDegreesCelcius;
double dailyTemperatureMeasureDecrementInDegreesCelcius;

// possible solution: shorten name
double temperatureIncrement_Celcius;
double temperatureDecrement_Celcius;


// what can you expect to find in one of these?
string productInfo;
string measuredData;
int value;
```

Los Angeles Times

Mars Probe Lost Due to Simple Math Error

BY ROBERT LEE HOTZ
OCT. 1, 1999 12 AM PT

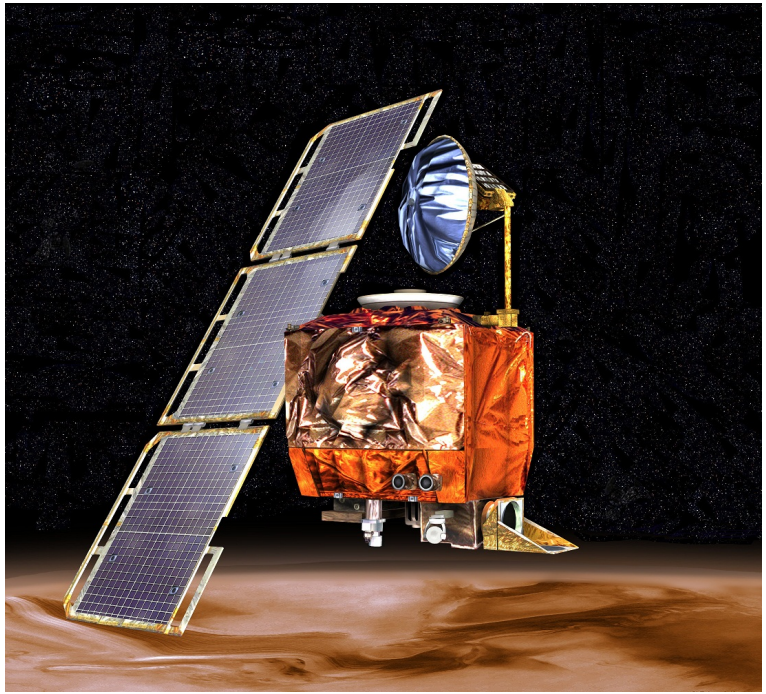
TIMES SCIENCE WRITER

NASA lost its $125-million Mars Climate Orbiter because spacecraft engineers failed to convert from English to metric measurements when exchanging vital data before the craft was launched, space agency officials said Thursday.

A navigation team at the Jet Propulsion Laboratory used the metric system of millimeters and meters in its calculations, while Lockheed Martin Astronautics in Denver, which designed and built the spacecraft, provided crucial acceleration data in the English system of inches, feet and pounds.

As a result, JPL engineers mistook acceleration readings measured in English units of pound-seconds for a metric measure of force called newton-seconds.

In a sense, the spacecraft was lost in translation.



```
double acceleration = getAcceleration();
```