UNIVERSITY OF TWENTE.

APPLICATION DEVELOPMENT

LECTURE 4: INTRODUCTION TO ARDUINO PROGRAMMING IN C++

```
class AppDev {
```


```
}
```

Part of SmartProducts

INTRODUCTION

APPLICATION DEVELOPMENT

- Introduction to Arduino programming in C++
  (comparison with Java)
- Wheeled vehicles
- Assignment

Fjodor van Slooten
W241 *(Horst-wing West)*
f.vanslooten@utwente.nl

slides @ vanslooten.com/appdev

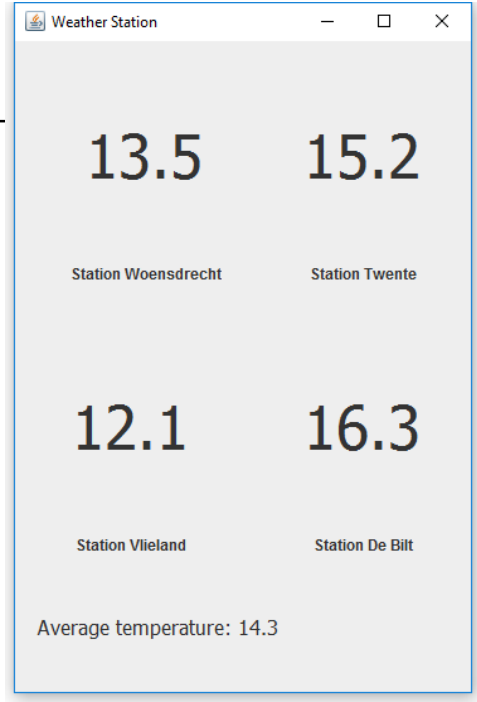
ASSIGNMENT 3

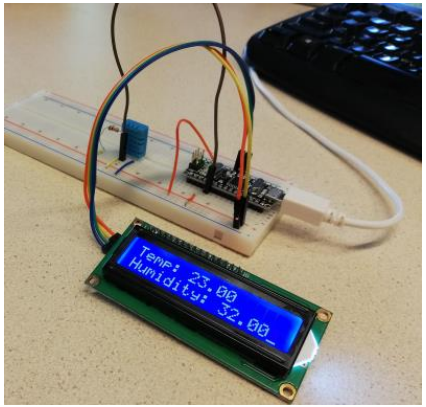
LAST SESSION

- Java assignment most of you did very well

Afternoon:

- Build Arduino circuit which displays temperature & humidity
- Connect it (with Java App/Blink)

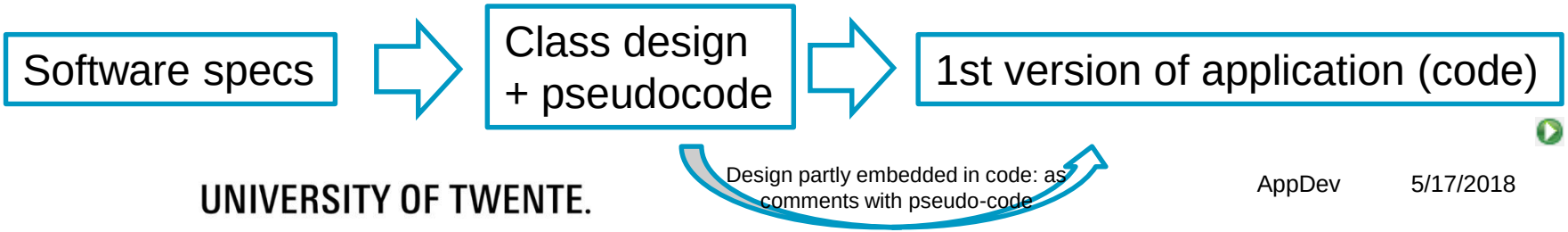




FROM SPECS, TO DESIGN, TO CODE

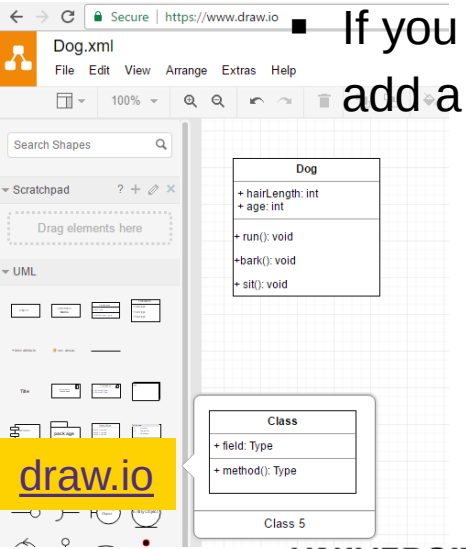
Head First: 1-4 ***Aan de slag met*:** 4.8-4.11, 6.1-6.4

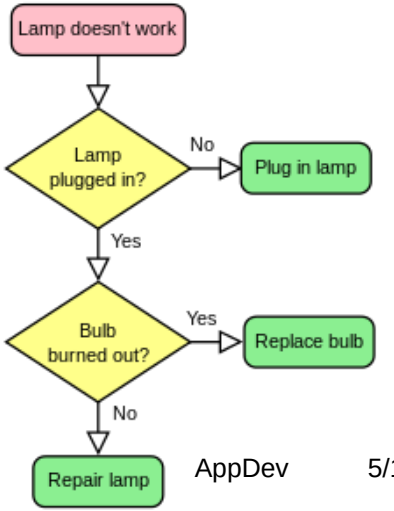
- How do I design an application? Check out previous presentations & assignments, read book
- Results of design-phase:
  - Pseudo code
  - Sketches
  - Diagrams (class-diagram, flowchart, …)



DIAGRAMS

- Can enhance the quality of your design
- If you have multiple classes, create a **class-diagram**
- If you have a complicated flow (of conditions, loops, etc.) add a **Flowchart**





FINAL PRODUCT? OR PROTOTYPE?

- Project is focused on prototype (although final product should be considered in design phase)
- @Application Development, design and code is restricted to prototype only
- This means, for the prototype you can:
  - Eliminate/simplify things you can not build
  - Simulate/demonstrate if necessary
- You may show design sketches of final product, as clarification

convince & attract

A prototype demonstrates design to client/ potential customers. And/or you use it for usability tests (goal: prove it can work).

ARDUINO: BASED ON C, C++

DIFFERENCES WITH JAVA

- You can use functions without a class:

You may consider a function is ‘a method without a class’

Standard functions `setup()` and `loop()`

```
void setup(){
  Serial.begin(9600);
}

void loop() {
  int sensorValue = readSensor();
  delay(500);
}

int readSensor(){
  int result = analogRead(A0);
  return result;
}
```

- Group code, use (call) from multiple locations

Added function `readSensor()`

arduino.cc/en/Reference/FunctionDeclaration

A CLASS IN C++

DIFFERENCES WITH JAVA

C++ is used in Arduino IDE

Java:

Dog.java

```
class Dog {
   // properties:
   int hairLength;
   int age;

   // methods:
   public void run() {
   }
   public void bark() {
   }
   public void sit() {
   }
}
```

C++:

Dog.h

```
class Dog {
   // properties:
   int hairLength;
   int age;

   // methods declaration:
public:
   void run();
   void bark();
   void sit();
};
```

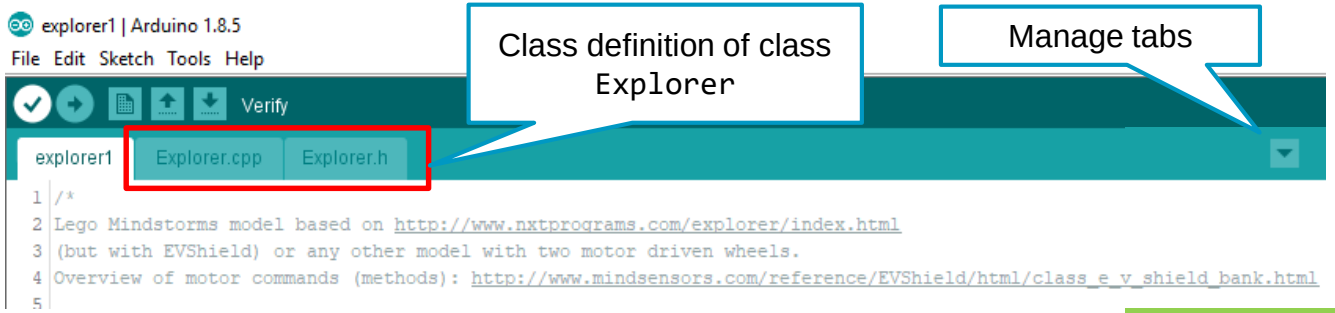
Declaration

Dog.cpp

```
// methods definition:
void Dog::run() {
}

void Dog::bark() {
}

void Dog::sit() {
}
```

Method definition

C++
COMPARED TO JAVA



Explorer.h

```
class Explorer {
  // class variables:
private:
  // pointers to objects:
  EVShield * evshield;
  NewPing * sonar;
  EVs_NXTTouch * touch;
...
void drive(int distance = 0);
...
```

Explorer.cpp

```
void Explorer::drive(int distance = 0) { // distance = 0 means drive unlimited

if (distance == 0) {
  Serial.println("drive unlimited");
  evshield->bank_a.motorRunUnlimited( SH_Motor_Both, motor_direction, speed );
  isDriving = true;
  evshield->ledSetRGB(0, 255, 0); // led green (indicates we're driving)
}
```

-> is pointer reference: call method **ledSetRGB()** of object *evshield* (you might be used to use a . here)

STRING

A-STRING-OF-CHARACTERS

"Hello, I am a String"

- String is not a standard Class in C++
- You must include Arduino.h:

```
#include <Arduino.h>

class Product {
private:
// class variables:
   String name;
   unsigned int price;
...
}
```

CLASS A USES CLASS B

This does not work: "class name does not name a type" (chicken-and-egg situation)

```
#include <B.h>

class A {
 public:
    A(int id);
 private:
    int id;
    B * objectB;
};
```

```
#include <A.h>

class B {
 public:
    B(int id);
 private:
    int id;
    A * objectA;
};
```

Solution: use *forward declaration:*

```
class B;

class A {
 public:
    A(int id);
 private:
    int id;
    B * objectB;
};
```

```
class A;

class B {
 public:
    B(int id);
 private:
    int id;
    A * objectA;
};
```

Google the error! > stackoverflow.com/questions/3608305/class-name-does-not-name-a-type-in-c

MATH

- PI
- abs()

Java:

```
double circumference = WHEEL_DIAM * Math.PI;
unsigned int degrees = (Math.abs(distance)/circumference) * 360;
```

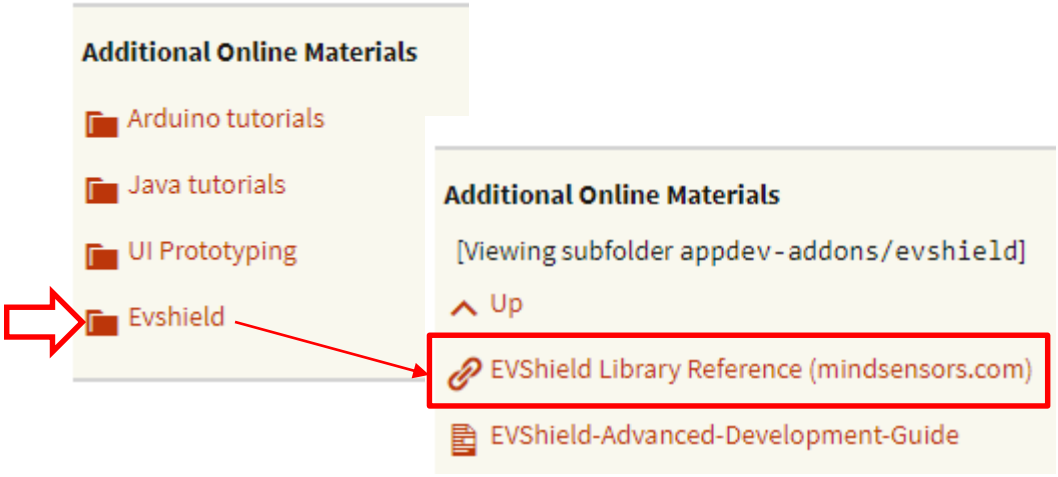
C++:

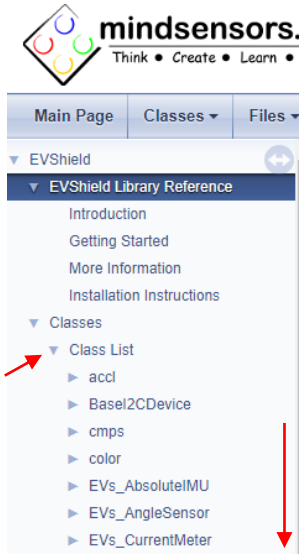
```
double circumference = WHEEL_DIAM * PI;
unsigned int degrees = (abs(distance)/circumference) * 360;
```

arduino.cc/en/Math/H

ACCESS EVSHIELD LIBRARY REFERENCE

Motor… commands?





scroll down until "**EVShieldBank**", click that

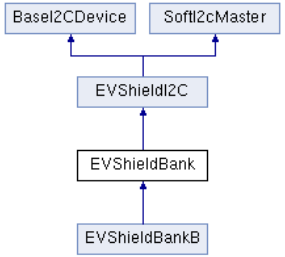
This is also available on your own computer:
Documents\Arduino\libraries\EVShield\html\ index.html

EVShieldBank Class Reference

This class defines methods for the **EVShield** Bank(s). More...

```
#include <EVShield.h>
```

Inheritance diagram for EVShieldBank:



Public Member Functions

	EVShieldBank (uint8_t i2c_address=**SH_Bank_A**)
int	**evshieldGetBatteryVoltage** ()
int	**nxshieldGetBatteryVoltage** ()
uint8_t	**EVShieldIssueCommand** (char command)
bool	**motorSetEncoderTarget** (**SH_Motor** which_motor, long target)
long	**motorGetEncoderTarget** (**SH_Motor** which_motor)
bool	**motorSetSpeed** (**SH_Motor** which_motor, int speed)
int8_t	**motorGetSpeed** (**SH_Motor** which_motor)
bool	**motorSetTimeToRun** (**SH_Motor** which_motor, int seconds)
uint8_t	**motorGetTimeToRun** (**SH_Motor** which_motor)

Methods of **EVSHieldBank** Class, a lot motor-related

CALCULATE MOTOR DEGREES

TO DRIVE A GIVEN DISTANCE

Let's drive …m

$$C_w = D_w \times \pi$$

$$degrees = \frac{distance}{C_w} \times 360$$

Drive a given distance (variable `distance` is parameter)

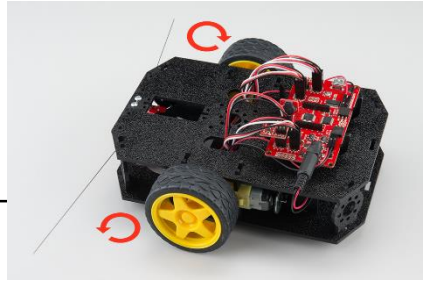
```
double circumference = WHEEL_DIAM * PI;
unsigned int degrees = abs(distance)/circumference * 360;

if (distance>0) { // positive distance, so just go in the current direction
   evshield->bank_a.motorRunDegrees(SH_Motor_Both, motor_direction, speed, degrees,
      SH_Completion_Wait_For, SH_Next_Action_Float );
}
else { // negative distance, so go other direction:
   SH_Direction use_motor_direction = SH_Direction_Forward;
   if (motor_direction==SH_Direction_Forward) use_motor_direction=SH_Direction_Reverse;

   evshield->bank_a.motorRunDegrees(SH_Motor_Both, use_motor_direction, speed, degrees,
      SH_Completion_Wait_For, SH_Next_Action_Float );
}
```

Run the motor for the calculated amount of degrees

Download example **`evshield_drive_1m.ino`** or see method **`Explorer::drive(int distance = 0)`** in code of assignment

TURN ANGLE

SPIN WHEELS IN OPPOSITE DIRECTION AT THE SAME TIME



= point turn

Turn with a given angle (variable `angle` is parameter)

Calculate degrees depending on wheel diameter and trackwidth
See next slide

Important: commands should run at same time! How? For first command use **`SH_Completion_Dont_Wait`**

```
unsigned int degrees = abs(angle) * (TRACKWIDTH / WHEEL_DIAM);

if (angle>0) {
   evshield->bank_a.motorRunDegrees(SH_Motor_1, SH_Direction_Forward, speed, degrees,
      SH_Completion_Dont_Wait, SH_Next_Action_Float);
   evshield->bank_a.motorRunDegrees(SH_Motor_2, SH_Direction_Reverse, speed, degrees,
      SH_Completion_Wait_For, SH_Next_Action_Float);
}
else {
   evshield->bank_a.motorRunDegrees(SH_Motor_2, SH_Direction_Forward, speed, degrees,
      SH_Completion_Dont_Wait, SH_Next_Action_Float);
   evshield->bank_a.motorRunDegrees(SH_Motor_1, SH_Direction_Reverse, speed, degrees,
      SH_Completion_Wait_For, SH_Next_Action_Float);
}
```

See method **`Explorer::turn(int angle)`** in code of assignment

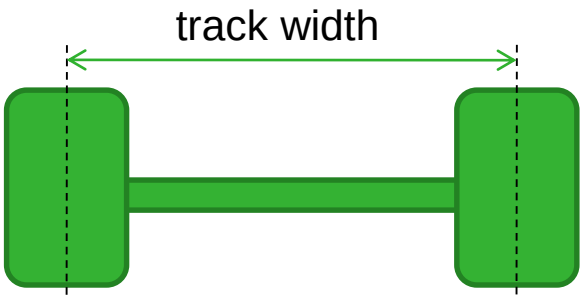
WHEELED VEHICLES

GIVEN WITH ASSIGNMENT: EXPLORER CLASS

```
class Explorer {
```

- Class definition in **`Explorer.h`**
- Parameters: wheel diameter, track width, motors

```
#define WHEEL_DIAM 4.96 // wheel diameter in cm
#define TRACKWIDTH 14   // track width in cm
```



$$Rot = n \times \frac{W_t}{D_w}$$

Make a 90 degrees turn: wheel diameter=4.96, trackwidth=14:

$$Rot = 90 \times \frac{14}{4.96} = 254$$

Rot = rotate motor in degrees
n = degree of turn
W_t = track width
D_w = diameter of wheel

Source for calculation

DRIVING & STEERING

WHEELED VEHICLES

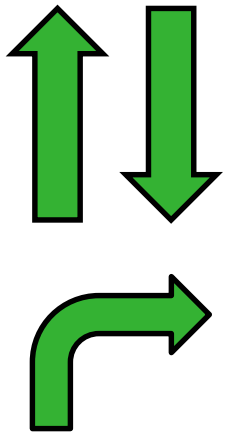
```
class Explorer {
```

Declaration in `Explorer.h`, Definition in `Explorer.cpp`

```
// driving related methods:
void reverseDirection();
void drive(int distance = 0);
void stop();
void turn(int angle);
```

Use (method calls)

```
// methods calls:
reverseDirection();
drive(); // drive unlimited (in current direction)
drive(200);
drive(-30);
stop();
turn(90); // turn right
turn(-90); // turn left
```



TRIP DATA

WHEELED VEHICLES

```
class Explorer {
```

Calculate travelled distance:

- Get number of degrees motor has turned (since its reset):

```
evshield->bank_a.motorGetEncoderPosition(SH_Motor_1)
```

- Calculate:

```
double circumference = WHEEL_DIAM * PI;
double m = abs(evshield->bank_a.motorGetEncoderPosition(SH_Motor_1))/360*circumference/100;
Serial.print("travelled distance: ");
Serial.println(m);
```

Blue part: number of rotations

Divide by 100 to get meters

what will go wrong here?
if we drive 1m forward, then 1m backward, travelled distance will be 0…

…what would be a solution?

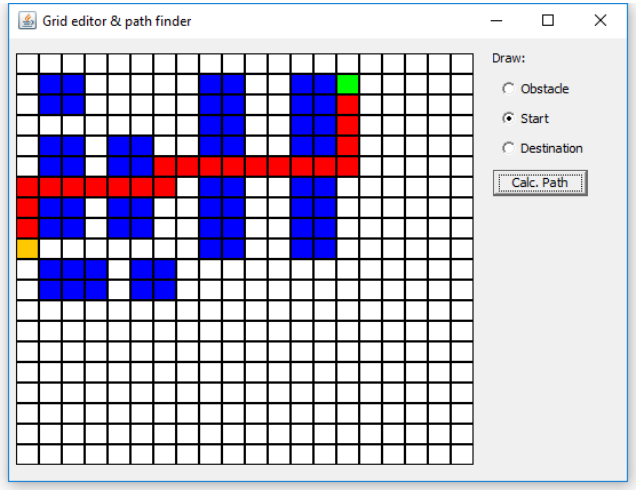
PATH-FINDING
WHEELED VEHICLES

More on this topic in session #6. Assignment 6 will be 'hands-on' path-finding.

Reading:

- Wikipedia: Pathfinding
- Wikipedia: A* search algorithm
- Introduction to A* pathfinding
- Path finding using A* Algorithm: Java Example
- Draw a grid-based representation of a room
  (solution at bottom using PixelPainter class)

Assignment 6 will be a combination of these 2



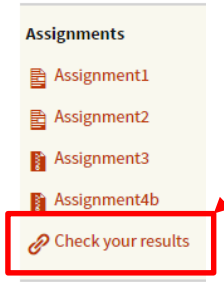
ASSIGNMENT #4

This afternoon: teacher available for help with project & Catch-up session if you missed a practical session

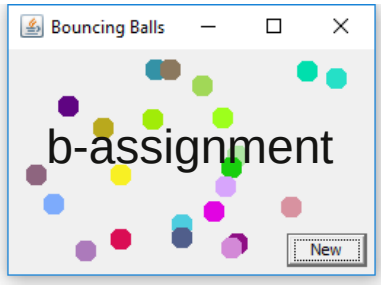
- “Program the Explorer robot”
- Read intro carefully, review all points in the checklist
- Sit together with project-group, do assignment in couples of 2 students!

Checkpoint

- Check assignments results



a or b?
Assignments which have 'a' / 'b' variant:
a-version = Arduino programming
b-version = Java programming (no Lego/Arduino required).
You have to do only one (a or b).



Slides, assignments etc @ vanslooten.com/appdev