



# PROGRAMMING IMPLEMENTATION WORKSHOP

IEEE CITY ROBOTICS SOCIETY



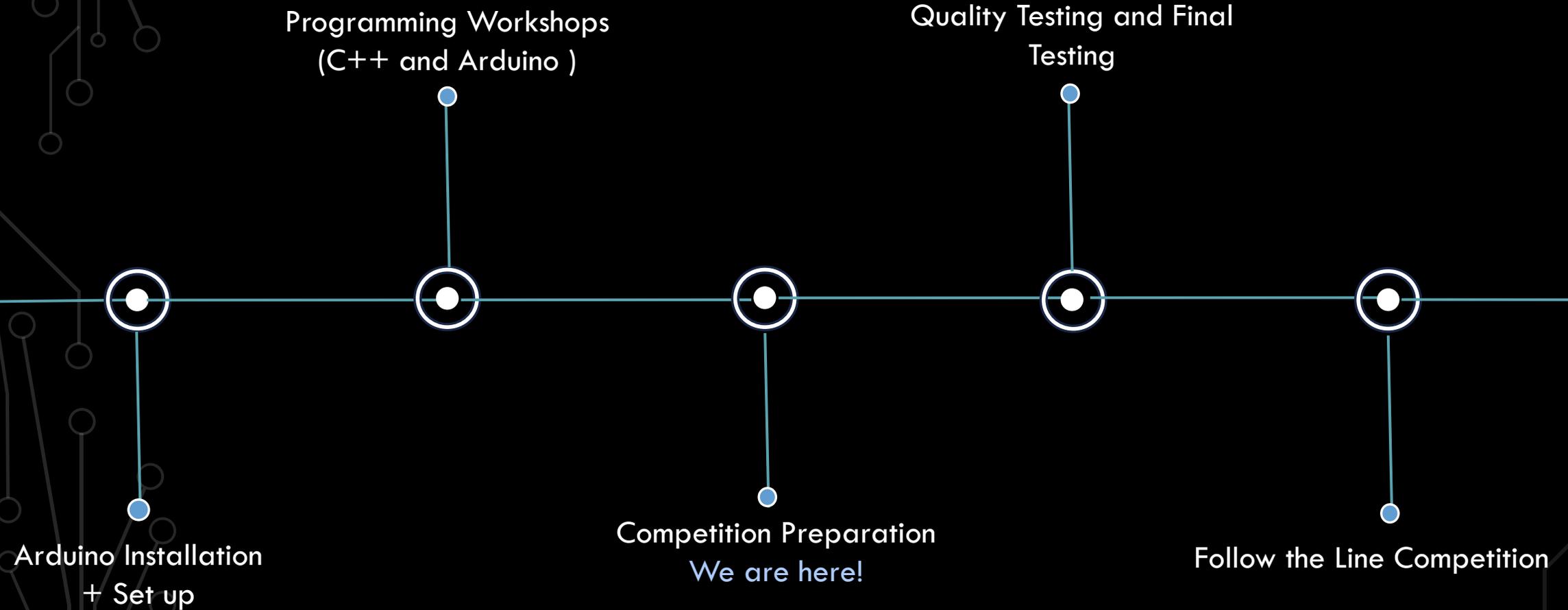
IEEE CITY  
ROBOTICS SOCIETY

# Today's Agenda

- Announcements
- Recap of Data Types
- Functions
- How to get the robot to move?
- Loops
- Data Reading using Sensors.
- Debugging
- Things to think about.



## Timeline for our upcoming events



# ANNOUNCEMENTS

- There will be an optional workshop taking place this Friday from 3pm to 5pm in room C302.
- There will only be 3 more workshops before the competition including this one.
- Please note that we will not be teaching any content during the Friday's workshop.
- If there is any event, you want us to hold, please come and talk to either us or our student reps: Andrei and Yousuf.

# MORE INFORMATION ABOUT THE COMPETITION

- 16 teams will be participating in this event.
- We are going to be using a bracket system to determine the winner.
- There will initial rounds, semi-finals and then finals.
- 2 seen tracks and one unseen track.
- Top 3 will receive a trophy for their team and some other prizes.
- All participants will receive a certificate of participation though.
- Biggest Robotic Showdown of the Year!

# RECAP ON DATA TYPES

## What are data types?

- Data types are the classification of a variable which determine which values it can hold and its respective operation.

## What does these include?

- **Integer** – Represent whole numbers without a decimal point. ( E.g. 5, 200, 1000, etc. )
- **Floating Point** – Represents a number with a decimal point or in exponential form (E.g. 3.14, 2.5,  $5.3e^3$ )
- **Character** – Represents individual characters which is usually denoted by single quotes (E.g. 'a', 'b', '9', '\$')
- **String** - This is a sequence of characters which is denoted by speech marks (E.g. 'bena', 'alan', 'hello')
- **Boolean** – These are variables that can either be assigned to true or false.
- **Array** – This is a data structure that represents a collection of values of the same type and is accessible using an index.

# VARIABLE SETUP IN ARDUINO

*Note: Throughout this session, we will be slowly building up the starting code for the robot. If possible, please try to follow along.*

- To operate and make the robot, we are going to be using the sensors that are at the bottom of the robot.
- There are 3 sensors in total. ( left, middle and right)
- We are going to define these variables at the beginning of our code file.
- To define a global variable for the sensors in Arduino, we need to use **#define**.
- We are also going to be using the function **analogRead()** which reads the values from analog sensors.
- Inside the parameters of this function, we must define the **pin** which it is pointing to.

# FUNCTIONS:

## What are Functions?

- A function is a named block of code that performs a specific task or set of tasks.
- They can be called at any point in your code to carry out a specific function.
- When creating a new function, the data type must be defined.
- We will be programming some functions later this session using the Arduino Environment.

## Why are they useful?

- Help in organising the code.
- Helps avoid repetition and code length.
- Makes it easier to read your code.

# FUNCTIONS THAT ARE USED IN C++:

- You will see 2 functions when you launch Arduino - Void setup() and void loop().

## **Void setup():**

- Return type of this function is void.
- This means that it this function is not returning a value.
- The main reason of this function is to set up the sensors at the bottom of the robot.

## **Void Loop():**

- This is the main loop and where all the functions will get called.
- There is only one main loop in a program at a time.
- Your main code needs to go in here.

# FUNCTION SET-UP IN ARDUINO

*Note: Throughout this session, we will be slowly building up the starting code for the robot. If possible, please try to follow along.*

## **Void set-up():**

- As mentioned, this function is used to assign the sensors in our robot to a variable.
- We can then use this variable in our main code and reference it.
- We will also be using the in-built function called **Serial.begin()**
- Live demonstration.

# IN-BUILT FUNCTION IN ARDUINO

- Arduino Library has a lot in-built libraries and functions that you can use.

So far, we have looked at:

- **analogRead()** – reads the sensors.
- **#define** – which allows us to create a global variable.
- **Serial.begin()** which allows us to start the sensors and read the data.

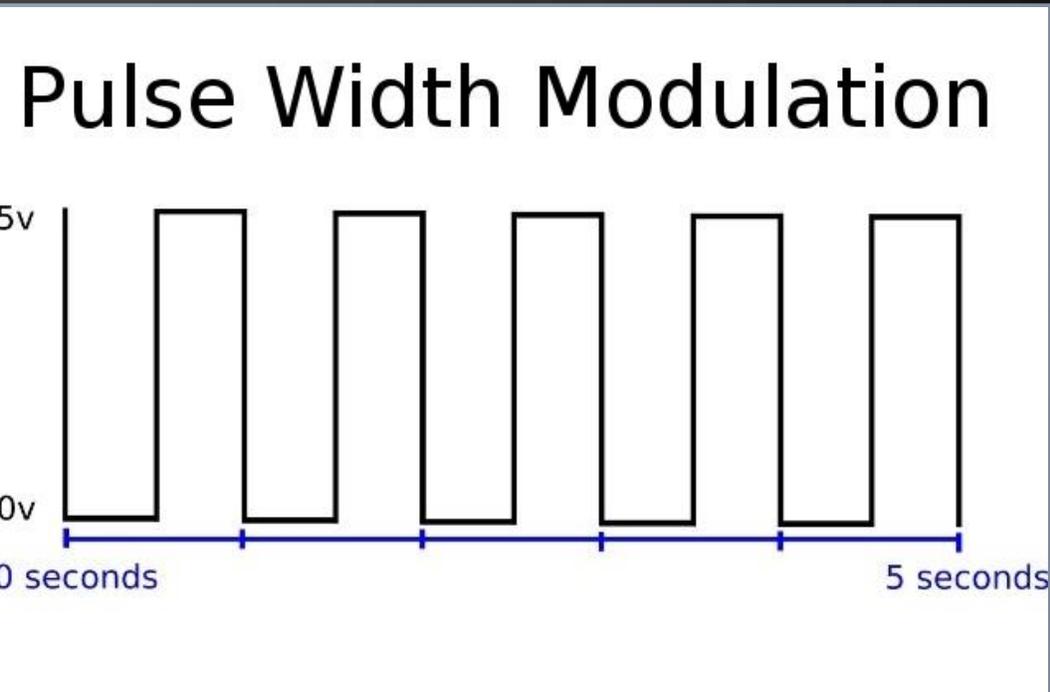
Other functions:

- **analogWrite(pin, value)** - used to generate analog output signals using Pulse Width Modulation (PWM).
- **digitalWrite()** - used to set the state of a digital pin to either HIGH or LOW.

# IN-BUILT FUNCTION IN ARDUINO – ANALOGWRITE()

- **analogWrite(pin, value)** - used to generate analog output signals using Pulse Width Modulation (PWM).
- The robot sends analogue signals through pin number.
- The value should be an integer between 0 and 255.

(Note: Think of this as driving speed of a car)



# IN-BUILT FUNCTION IN ARDUINO – DIGITALWRITE()

- **digitalWrite(pin, value)** – used to set the state of the digital pin.
- i.e., Used to define the speed.
- The State can be either **HIGH** or **LOW**
- If **HIGH** = Motor is turned on
- If **LOW** = Motor is turned off.

# IN-BUILT FUNCTION IN ARDUINO – DIGITALWRITE()

Each Motor has a different pin it relates to:

- **Pin 5 = Back right motor**
- **Pin 6 = Back left motor**
- **Pin 7 = Forward Right**
- **Pin 8 = Forward Left**

# KEY FUNCTIONS SET-UP IN ARDUINO:

We will start with explaining the robot manoeuvres.

- Remember, the robot will only do what you instruct the robot to do.

## Going forward:

- All motors must be turned on. (HIGH)

## Turning right:

- The front left motor must be turned on. (HIGH)
- The back left motor must be turned on. (HIGH)
- The front right and back right must be turned off. (LOW)

// can you try for the function turning left and going backwards?

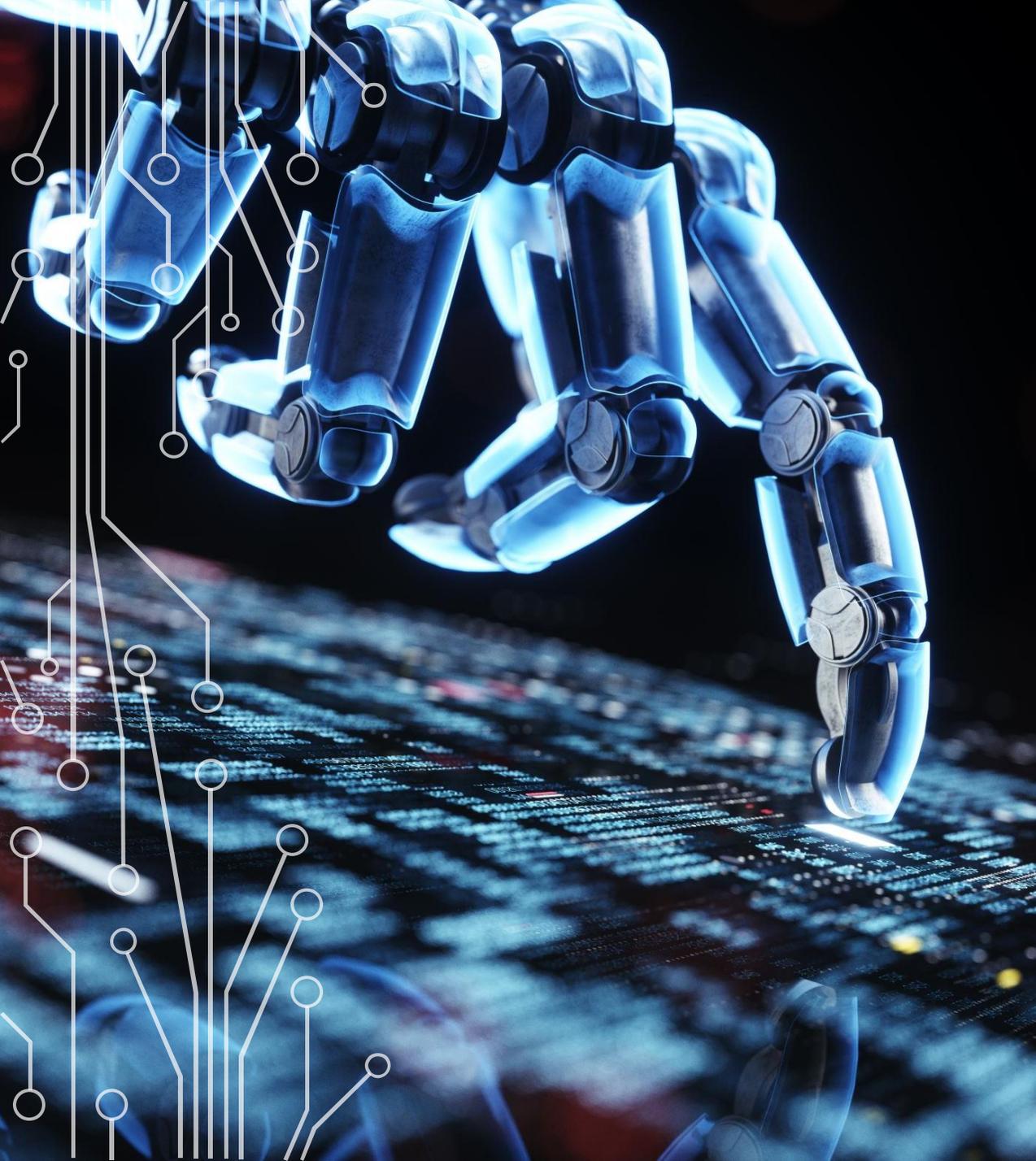
# KEY FUNCTIONS SET-UP IN ARDUINO:

## Going backward:

- The front right motor must be turned off.
- The front left motor must be turned off.
- The back left motor and the back right motor will be turned on.

## - Turning left:

- The front right motor must be turned on. (HIGH)
- The back right motor must be turned on. (HIGH)
- The front left and back left must be turned off. (LOW)



## HOW TO GET THE ROBOT TO MOVE?

To get the robot to move, you need to call your functions in the main void Loop().

- You will need to use Loops.
- Live demonstration.

# LOOPS:

## What are Loops?

- These are a control flow structure that allow a set of instructions to be repeated multiple times while the condition is met.

## There are several types of loops:

- **For loops**
- **Do while loops**
- **While loops**
- **If Loops.**

# FOR LOOPS:

```
1
2 void setup() {
3   // put your setup code here, to run once:
4   Serial.begin(9600);
5   for(int i=0;i<10;i++){
6     Serial.print("Value of i is: ");
7     Serial.println(i);
8   }
9
```

# WHILE LOOPS:

sketch\_nov20a.ino

```
1  int i=0;
2  void setup() {
3      // put your setup code here, to run once:
4      Serial.begin(9600);
5      while(i<10); {
6          Serial.print("Value of i is: ");
7          Serial.println(i);
8          i++;
9      }
10
```

## DO WHILE LOOPS:

```
1  int i=0;
2  void setup() {
3      // put your setup code here, to run once:
4      Serial.begin(9600);
5      do{
6          Serial.print("Value of i is: ");
7          Serial.println(i);
8          i++;
9      }
10     while(i<10);
11
```

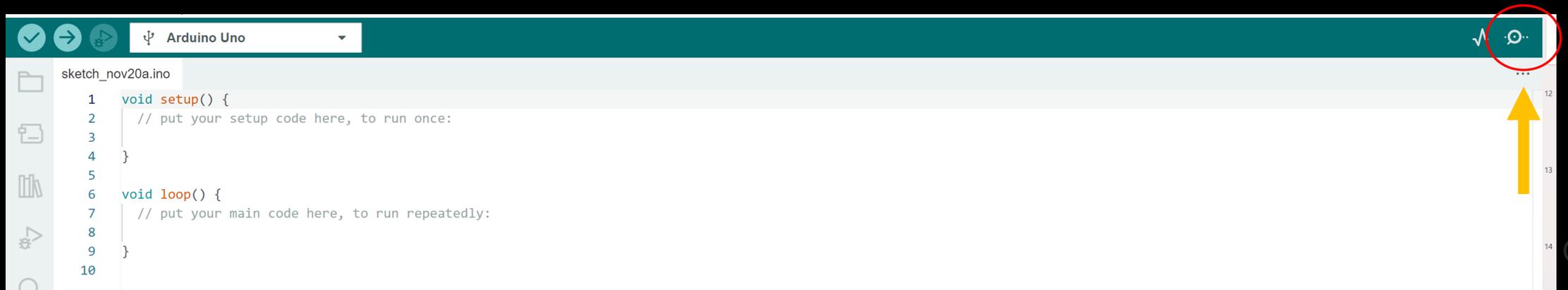
# IF LOOPS:

```
1   int Num1 = 20;
2   int Num2 = 5;
3   void setup() {
4       // put your setup code here, to run once:
5
6
7   }
8
9   void loop() {
10      // put your main code here, to run repeatedly:
11
12      if(Num1 > Num2){
13          Serial.println("Number 1 is greater than Number 2");
14      }
15      else{
16          Serial.println("Number 2 is greater than Number 1");
17      }
18
19  }
20
```

# HOW TO GET DATA READINGS FROM THE SENSORS?

To do this, we can call the function `Serial.Print()` - Used to print to the console terminal.

- Also, can be used for debugging purposes.
- For this, you need to call this function in the main loop. (Live demonstration).
- Then locate:



The screenshot shows the Arduino IDE interface. The top bar is teal and contains a dropdown menu set to "Arduino Uno". On the right side of the top bar, there is a red circle around a terminal icon. Below the top bar, the sketch editor shows the following code:

```
sketch_nov20a.ino
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
10
```

On the right side of the sketch editor, there is a vertical scrollbar with line numbers 12, 13, and 14 visible. A yellow arrow points upwards from the bottom of the scrollbar towards the terminal icon in the top bar.

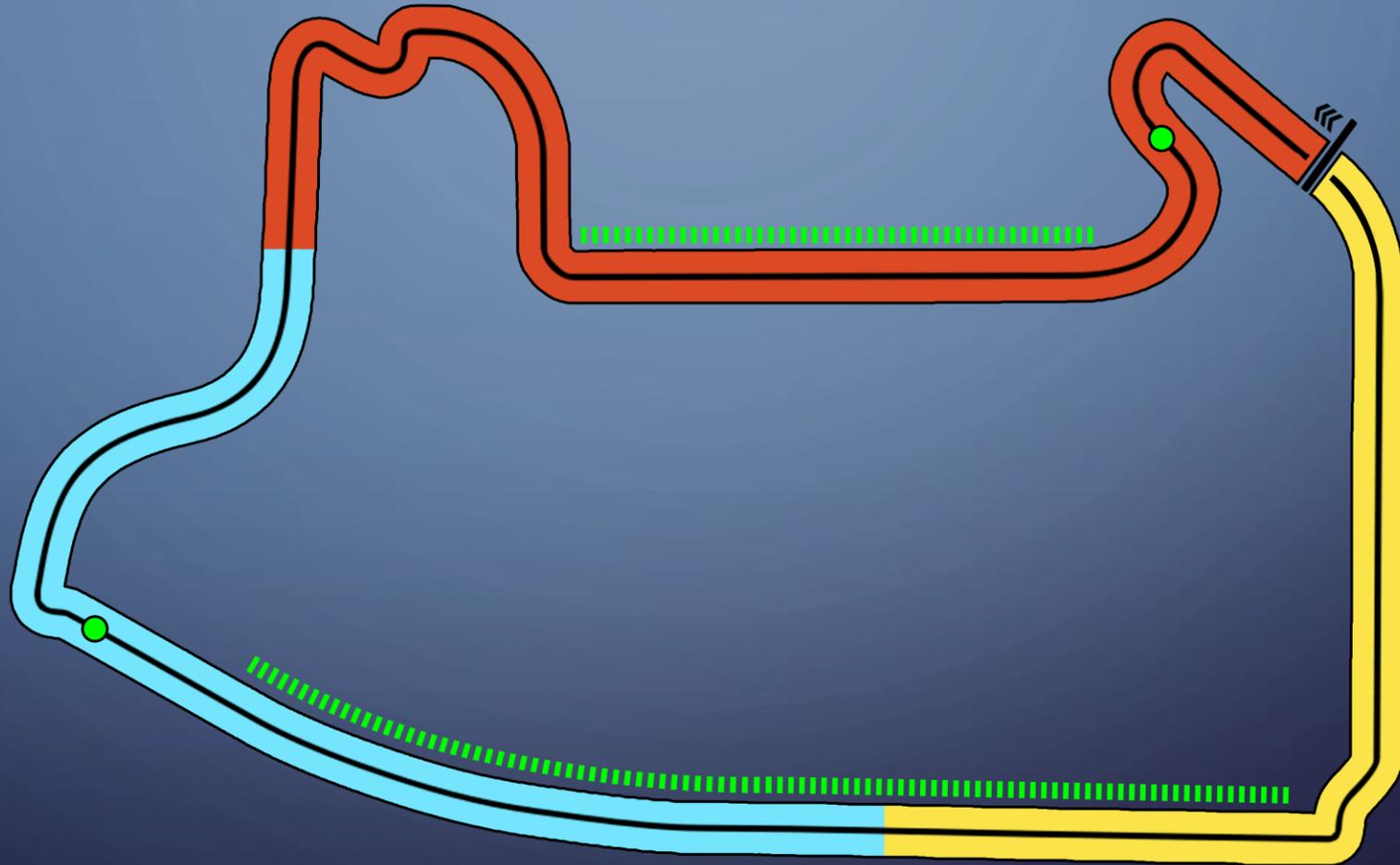
## FINAL GENERAL ADVICE FROM ME:

- Make sure you understand how the robot moves. If any doubt, please raise your hand during the session.
- We recommend taking data readings for each of the scenarios. ( E.g. dashed lines, straight lines, turning left etc. )
- Plan your strategy!
- You can use if statements to achieve this.
- Remember there will be awards for best code, best lap time, best design and best leader.
- But most of all, remember to have fun!

# TOURNAMENT STYLE

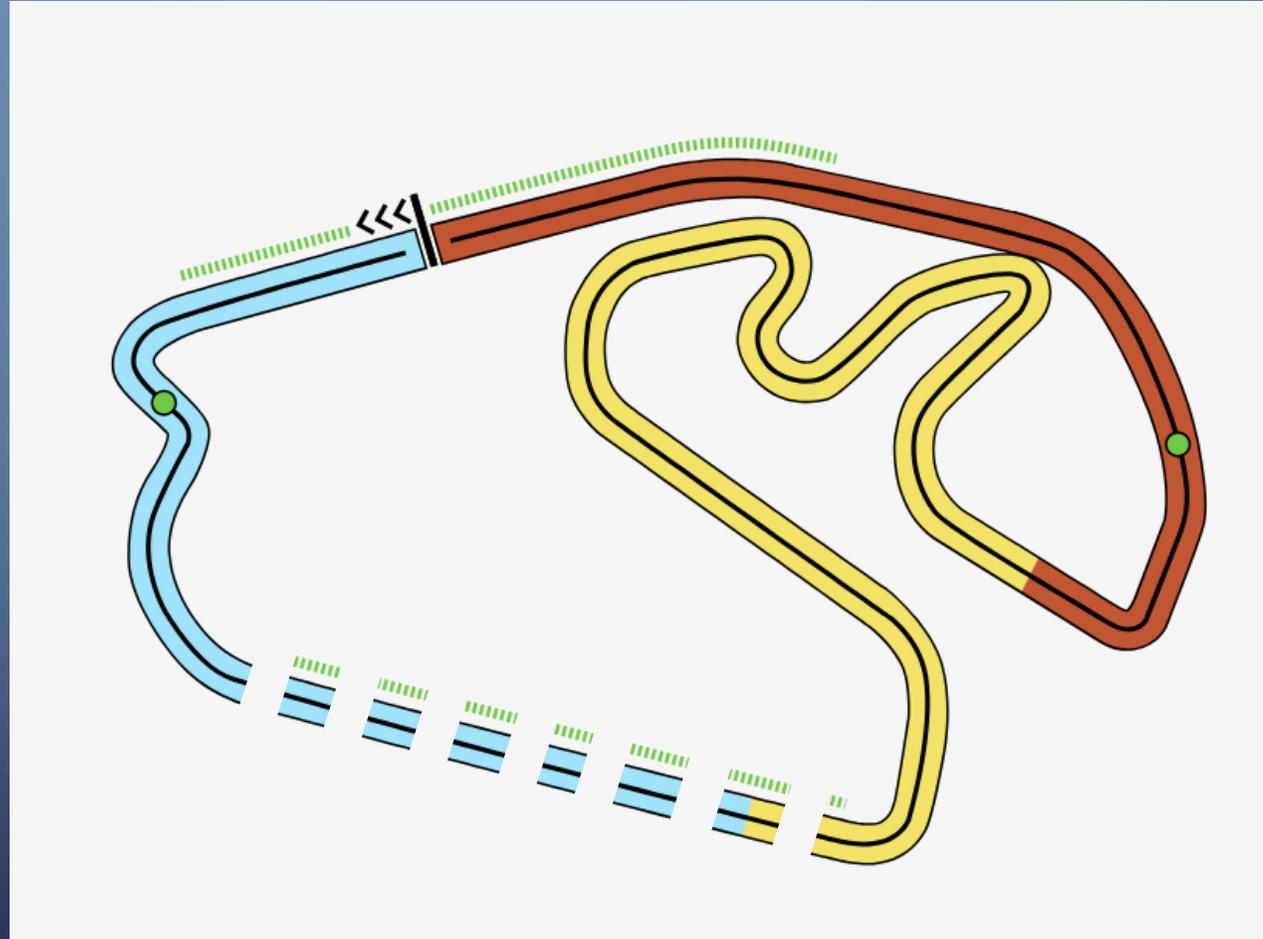
- Round 1: 16 Team Bracket Head 2 Head
- Round 2: Semi Finals! 8 Team Bracket Head 2 Head – different track!
- Round 3: Finals! Top 4 compete for the final round on a mystery unseen track!!
- 5 Minute pitstop break between each round to make modifications to code and robot (eg taking off / adding weight or components)
- Only the final state of your robot will count to the best design competition which will be put down to a popular vote at the end of the rounds.

# ROUND 1 TRACK\*



\*Adaptations as necessary

# ROUND 2 TRACK\*



\*Adaptations as necessary

## ROUND 3 TRACK\*

- MYSTERY TRACK FOR THE TOP 4 FINALISTS!
- WILL BE A CHALLENGE INCLUDING DASHED SECTIONS AND CROSSOVERS
- YOU MUST PREPARE CODE FOR ALL THREE TRACKS IF YOU WANT TO CHALLENGE FOR THE WIN!
- NO ONE SIZE FITS ALL
- USE YOUR 5 MIN PITSTOPS TO SWITCH OUT THE CODE!