

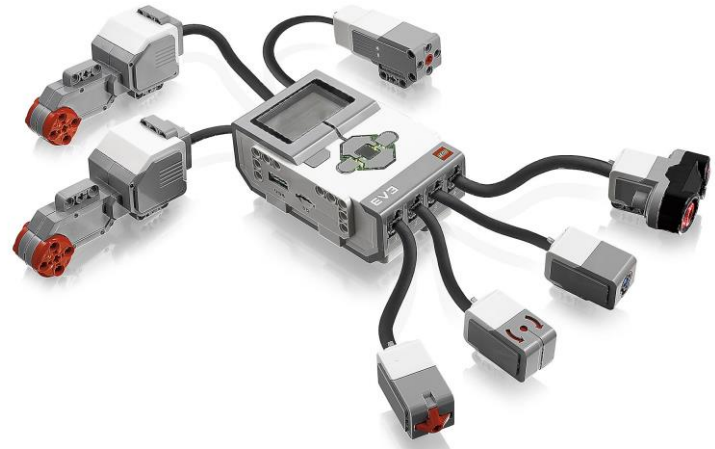


# UNDERSTANDING COMMUNICATION WITH A ROBOT? ACTIVITY (60 MINUTES)

1

# SUMMARY SO FAR

- The parts of a robot are
  - A computer that needs to be programmed (to make decisions)
  - Inputs (to 'sense' via sensors)
  - Outputs (to 'act', e.g., via motors)
- Our EV3 has all three of these, but in this activity we'll be using the computer and outputs.
- You'll make the EV3 robot run by creating a 'program' to tell the EV3 computer how to use its outputs.



# ACTIVITY PART I – FOR TEACHER

- **Split the students into groups of two**
- **Mark four points on the ground – say on the corners of a 5 foot square. Space the points farther apart to increase degree of difficulty.**
- **Have one person in each group close their eyes and have the other member direct the first around the room with verbal commands.**
- **The goal is to become the first group to have the member with his/her eyes closed touch all four points, following the RULES on the next slide.**

# ACTIVITY PART I – STUDENT RULES

## ○ Rules:

- The command giving partner can *only give commands that follow a specific format*: **The command must tell the partner to walk a certain number of steps in *one* direction.**
  - Example: Move three steps to your right.
  - Example of incorrect command: Move a little to your right then turn slightly left and walk forward.
    - Why above command is incorrect: First, it specifies movement in multiple directions, instead of only one direction. Also, command doesn't specify an exact number of steps to walk.
- ***NO COMMAND CAN BE GIVEN WHILE THE PARTNER IS IN MOTION!!*** Commands can only be given after previous commands have been executed.

# ACTIVITY PART II – FOR TEACHER

- **First you have to assemble the EV3 Robot using the documentation in the pdf “How to assemble an EV3 Taskbot”. This is in the attachment section in the TeachEngineering Activity “Understanding Communication with a Robot”.**
- **The process of assembling takes about 45 minutes for someone who is not familiar with LEGO kits.**
- **TWO OPTIONS – A) This assembly can either be performed outside of class with a group of selected students, OR B) can be performed in class by the student groups. If it is to be performed in class, this will take about 45 minutes.**
- **The slides that follow assume that the Taskbot has been assembled, and so move on to how to program it.**

# ACTIVITY PART II – FOR TEACHER

- Create a maze with wooden planks (or any other solid objects forming walls of the maze) – increase the size of the maze (add more turns) to increase difficulty.
- Use the PowerPoint slides that follow (slides 7-31) to instruct kids how to program using EV3 software.
- **Engineering Design Challenge - Using the LEGO EV3 software, ask students to create programs to get their taskbots through the maze.**
- Suggestion – Slides 7 through 31 can be printed and provided to the students so that they can use them to troubleshoot when they begin writing their programs.

# GETTING STARTED WITH LEGO EV3 MINDSTORMS SOFTWARE

There's more to the program than what can be found here, but the slides that follow can get you started and give you enough information to complete this activity

## Engineering Design Challenge

Using the LEGO EV3 software, create a program to get your taskbot through the maze.

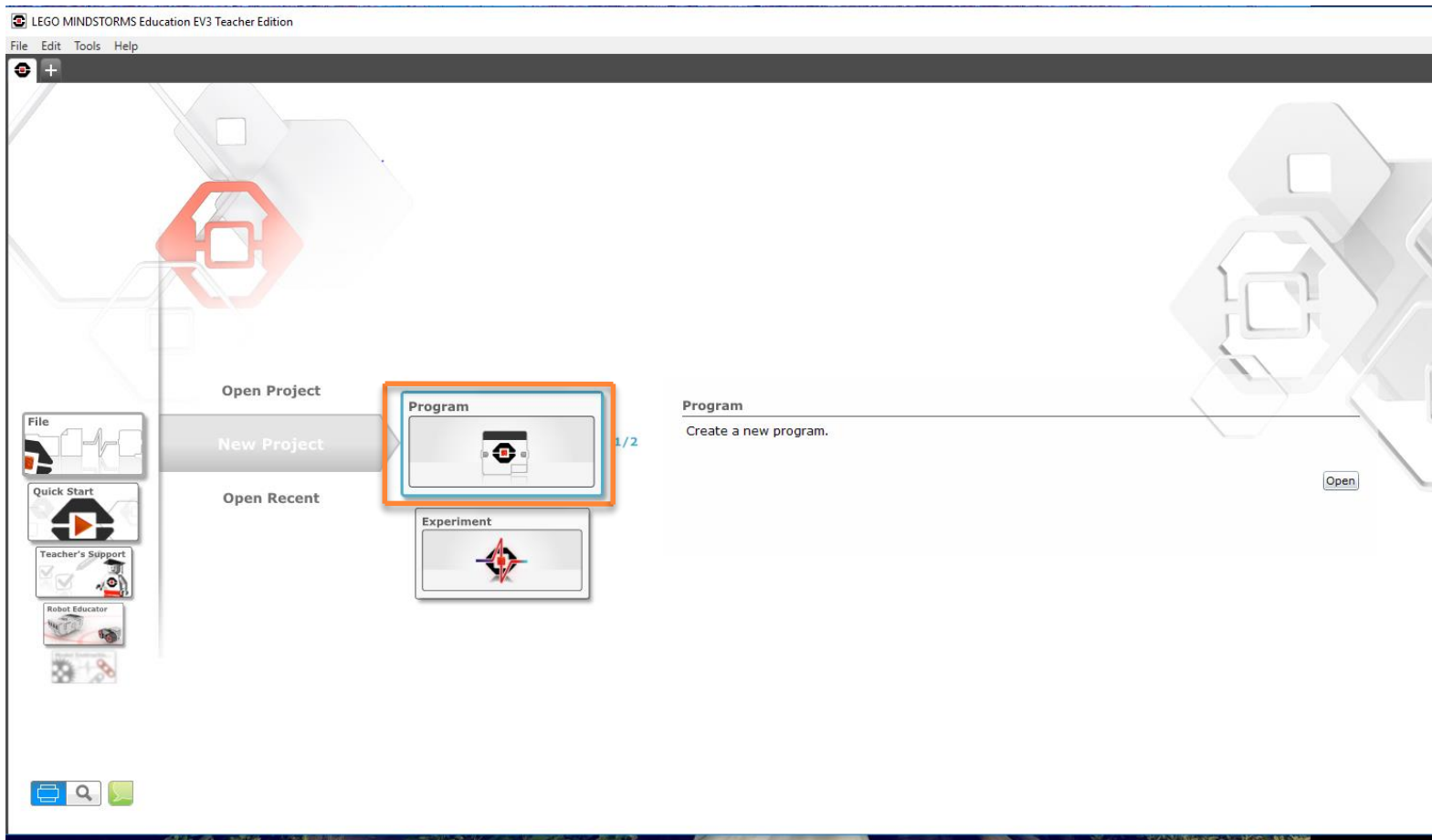
# STARTING A PROGRAM

- The EV3 kit comes with a software disk that you have to install on your computer (both Windows and Mac version are available). We assume you have installed it.
- When you open the EV3 MINDSTORMS software, you should encounter a screen that looks like this.





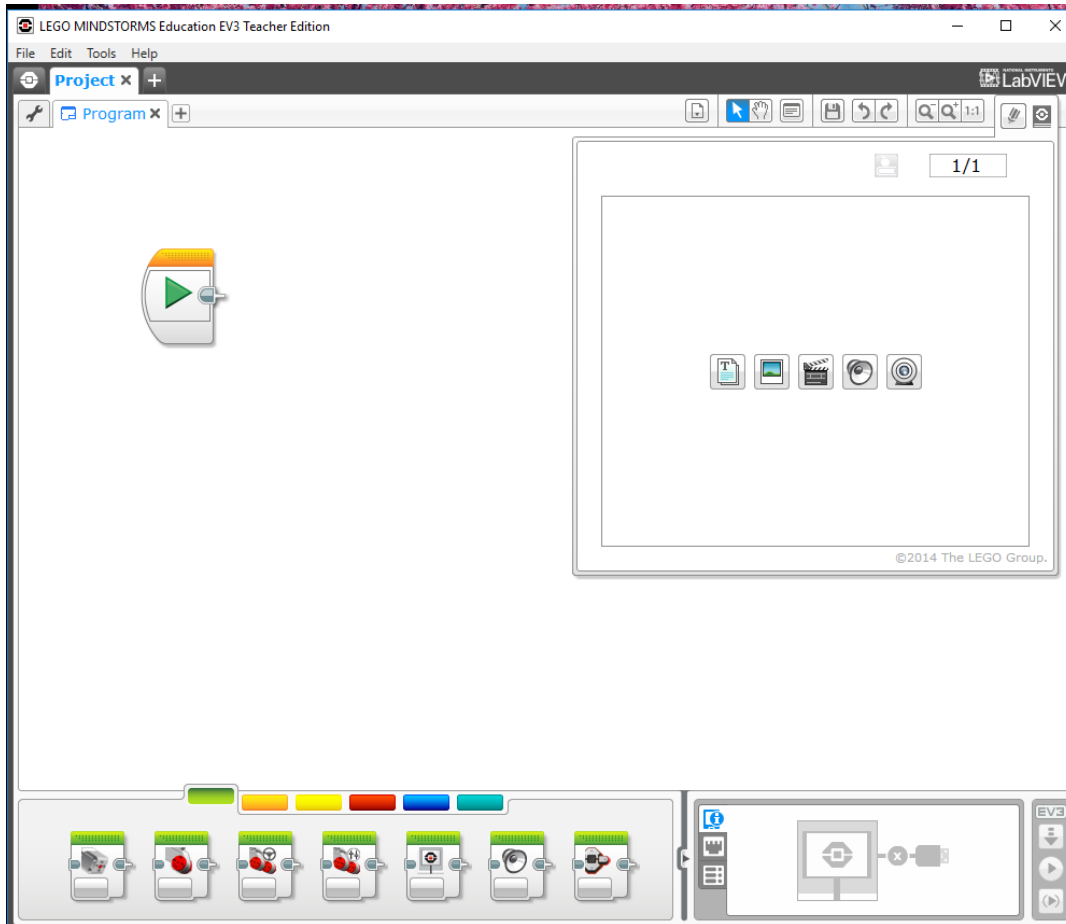
# STARTING A PROGRAM – CONTD.



On this screen, press “New Project >>> Program”.

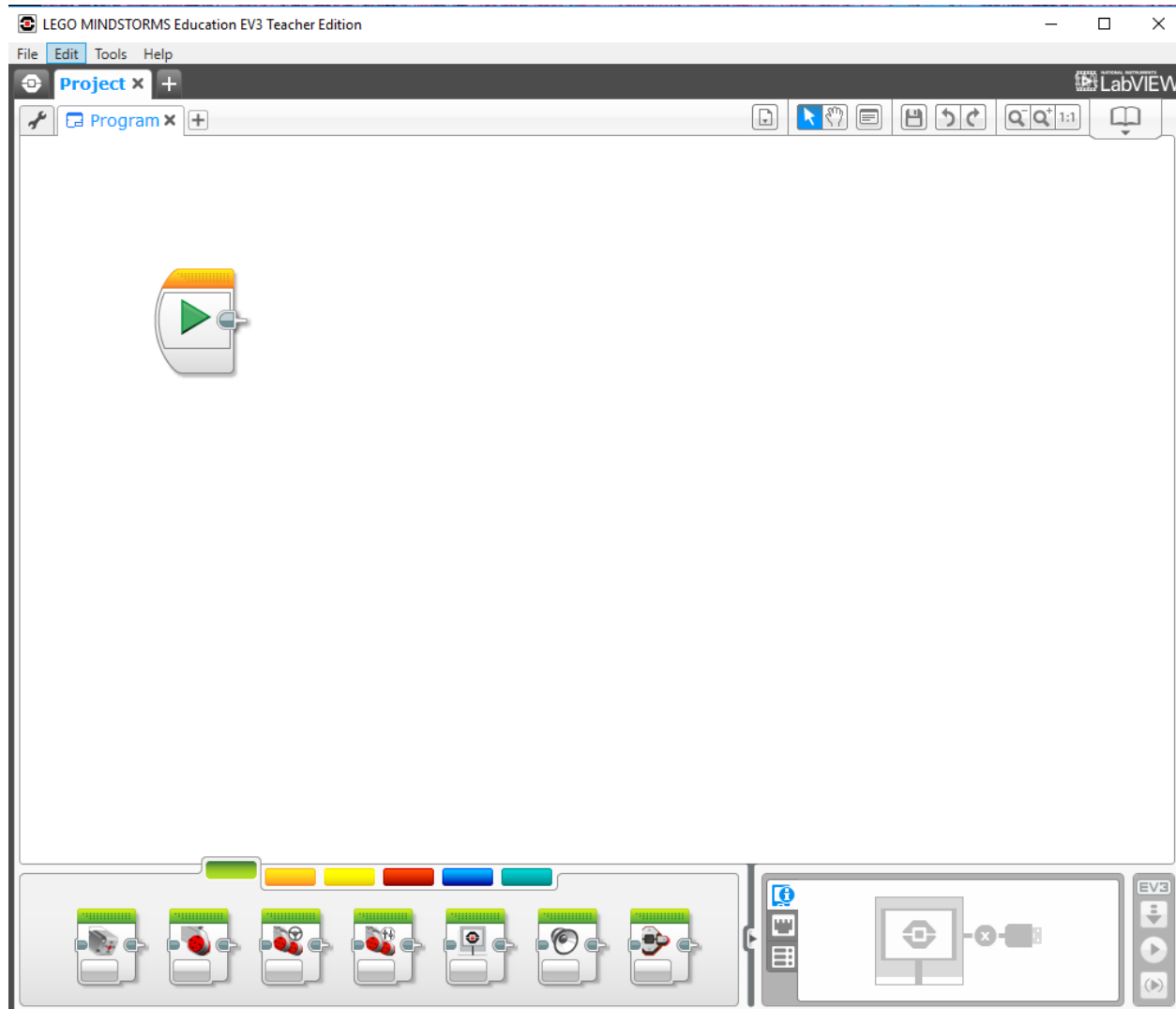
# STARTING A PROGRAM – CONTD.

- Your screen should now look like this

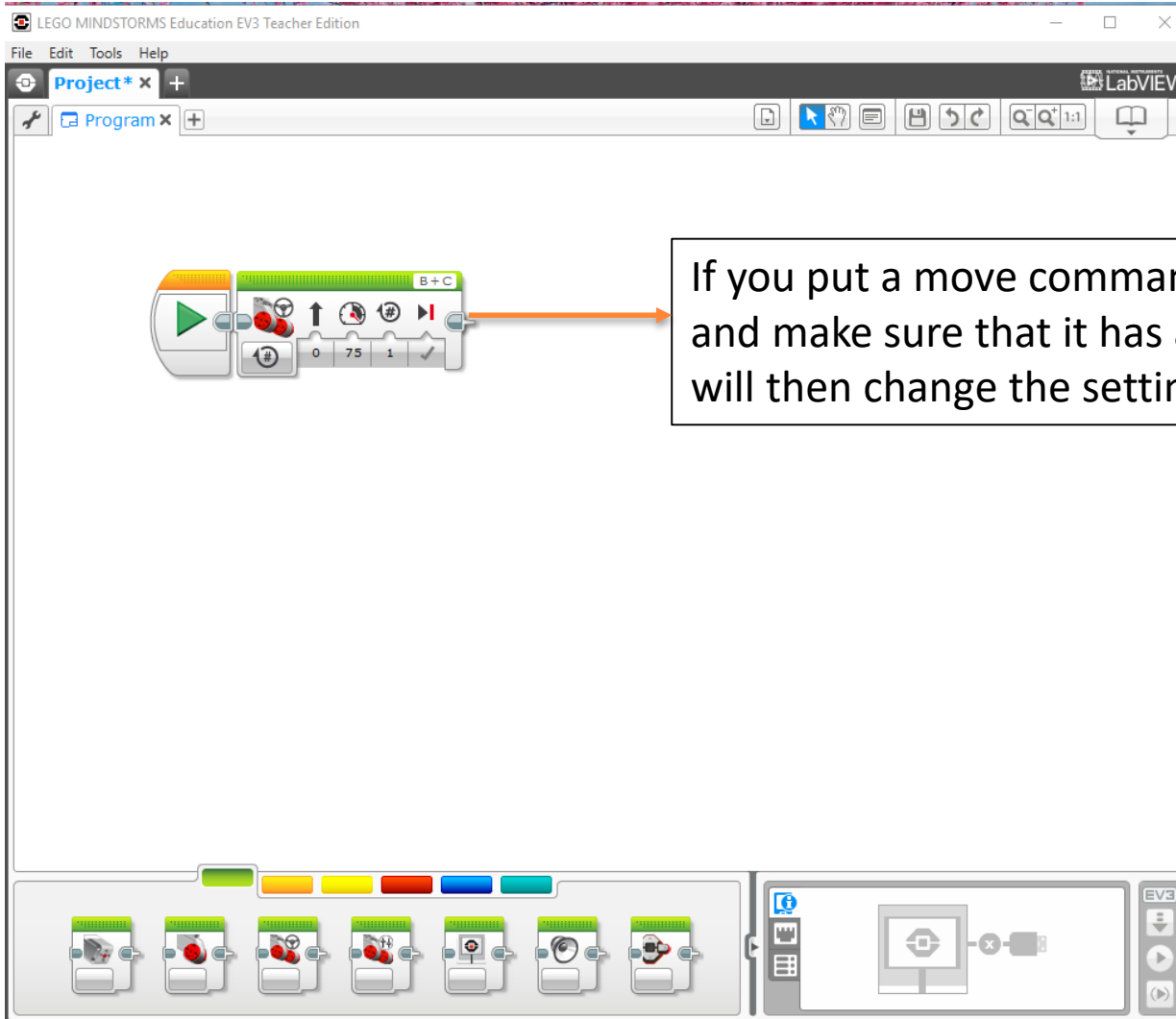


# TELLING THE ROBOT WHAT TO DO

## -- WRITING A 'PROGRAM'



- The command that we will be using in this activity is called the 'move' command



If you put a move command on to your program and make sure that it has a blue outline, you will then change the settings one by one.

# PORTS & STEERING

- First we'll discuss the ports – A, B, C



- Simply make sure that the port that your motor is connected to are the same ports that are checked in in the figure above (default programming settings are B to your right motor and C to your left motor. If you wire it that way programming will be much easier).
- *Helpful hint:* When navigating a maze or in any other tight spaces, it is better to use really **sharp** turns (move the sliding icon **all the way over to one side**), so that you avoid running into the walls of the maze or of what ever area you're in.

# POWER

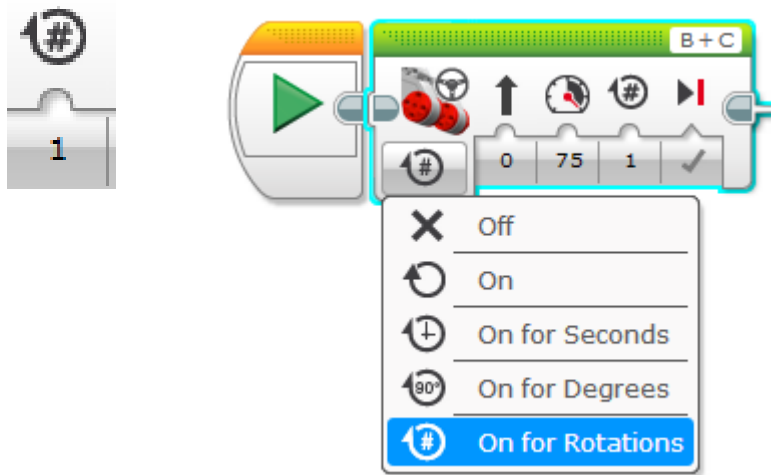
- This is the power box



- Everyone likes to run their power up to 100 to see how fast the robot moves but BE WARNED: lower power means better control, which makes it easier to navigate sharp turns and small areas in mazes!

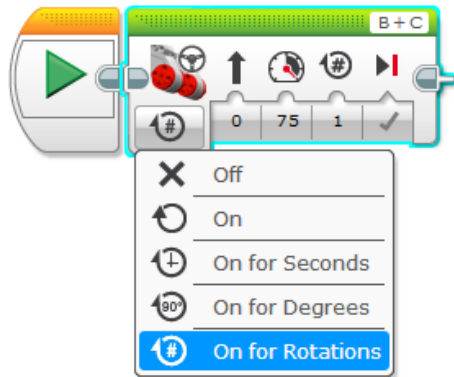
# DURATION

- This is the duration box. Click on the arrow next to the word rotations to see all of the duration options



- For now, ignore the unlimited option, since it only makes sense to use it with advanced programming cases.

# DURATION (CONT)



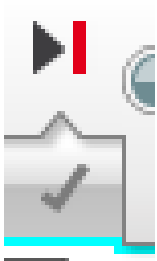
- Let's go over all of the other duration options
  - Seconds just tells your robot to use this specific command for a certain number of seconds.
  - Rotations tells your robot to use this specific move command until the wheels have turned a certain number of times (**number of rotations of the wheel – not of the robot!!**)
  - Degrees is the same thing as rotations, except instead of number of rotations it does the number of degrees that the wheel has turned (simply the number of rotations multiplied by 360).



# DURATION (CONT)

- BIGGEST MISTAKE PEOPLE MAKE:
  - Degrees only refers to the number of degrees the wheels of the robot will turn, not the number of degrees that the robot itself will turn!
    - That is, if you tell the robot to turn right for 90 degrees it will NOT make a 90 degree turn!!!
- Other important information
  - It's a good idea just to use rotations to determine the duration of a move command. It is the easiest to use since you'll know exactly how far the robot will go using that one command.

- Finally, this last command simply lets you choose what the robot does after the command is over (assuming it is the last command in your program). Your robot will either completely stop (brake), or it will keep rolling until friction stops it (coast)



# SOME HINTS

- Each different part of the maze needs its own move command (each straight part should have its own command and each move should have its own move), so start by counting the total number of move commands you need.
- A right angle turn is best done by moving the sliding bar for the turn of your move command all the way over and then setting the duration for 0.5 rotations (you may need to do 0.4 or 0.6).

# THE CHALLENGE

Using the LEGO EV3 software, write a program to get their robots through the maze



## SAMPLE PROGRAM (CONT.)

- To determine the right number of rotations for the straight portions of the maze, you need to use trial and error. Just remember to have the kids use 'rotations' (of wheels) for their duration, as it is the most consistent and easiest to work with.



## PRE/POST-ASSESSMENT SHEET - What is a robot?

1. Describe in one sentence below what you understand by the term 'robot'

**A robot is a machine that gathers information about its environment (senses) and uses that information (thinks) to follow instructions to do work (acts).**

2. What are the main parts of a robot?

**Computer (to make decisions), Input ports (connected to sensors), and Outputs (connected to motors, for example).**

3. What do you need to do to make a robot move?

**They program it using software telling it precisely what to do, step by step.**