InPython Course

CoyoteLearner

Basic Teacher Notes

You do not have to follow any of the advice given in this document. You may choose to follow it in your own way to take advantage of this learning resource. It is entirely up to you.

What you will need:

- You need a projector and good speakers. Alternatively, individual headphones for each computer and a good Internet connection. Please note that if you choose the headset option and you have a class of students, your Internet connection should be really good to cope with all the streaming videos.
- Every student needs to have a computer or a laptop (Not a tablet)
- Internet connection is required

Initially, it is very important to emphasise that the course is designed as e-learning. That is, the students may have access from home and may work relatively independently.

You will also note from the first lesson that a single view of each video is not enough for a student to master the curriculum. Each student, with their own distinct personality, has a personal learning path that differs from his or her classmates. One student may be content to watch a video once while someone else will need a lot more. You should note that in a class that has limited time, re-viewing videos is not possible for 2 reasons:

1. *You will not have enough time.*

2. Some students will get bored.

So access from home is essential. One of your roles is to support e-learning. You have to urge students to work at home. You need to help them get into a

rhythm and help increase their commitment to the course. As time passes, your role should become less important in their learning process.

Important notice:
You will certainly have noticed that Modules 4 and 6 refer to robotics and Edison robots, as well as some sub-modules and exercises of sections 8, 9 and 10.

The educational material presented by these modules mainly concerns robotics. Of course, Python is extensively used in these modules, but does not introduce new material elements.

This means that a teacher, if he or she does not have the Edison robotic system, can skip these modules.

Please first make sure that the students are already familiar with the platform. If not, you have to present it to them. They need to know how to connect to the system, how to get access to the course, points, levels, leader boards, how to reach you, and how to communicate with classmates. All the answers are in the "How to" courses which are free in the platform.

Curriculum

Currently 11 modules make up the 'InPython Course.'

Module 1 refers to the basic concepts of computer programming.

Module 2 is about getting started with IDLE editor and introduces the variables concept.

Module 3 is about in-built Python functions and expressions.

Module 4 is an introduction to robotics and the Edison Robot.

Module 5 introduces the object–oriented programming concepts by exploring the Turtle Python library.

Module 6 is about the Edison robot and its movements.

Module 7 refers to lists and iterations.

Module 8 introduces logical expressions and the selection structure concept.

Module 9 examines further the loops and especially the while statement.

Module 10 is about functions and flow charts and an introduction to event-driven programming.

Module 12 is an exercise where students have to program Edison Robot, and in this way to solve a "real-world" problem. Students are asked to solve the problem by analyzing it in smaller subproblems.

At the end of the course, students will have learnt the basics concepts of computer programming and robotics.

Curriculum

Modules	**Educational concepts**	**Suggested teaching hours**
Module 1	Computer programming theory	2 hours
Module 2	Editors and variables	2 hours
Module 3	Built in functions and expressions	2 hours
Module 4	Introduction to robotics and to the Edison Robot	2 hours
Module 5	Introduction to object-oriented programming	3 hours
Module 6	Edison robot movements	3 hours
Module 7	Lists and iterations	3 hours
Module 8	Basic Programming Concepts – Selection Structure	4 hours
Module 9	Basic Programming Concepts – Iteration Structure	6 hours
Module 10	Functions and flowcharts, introduction to event driven programming	6 hours
Module 12	Project: Parking Edison	5
		35 hours

Course plan

Module 1 – Introduction To Programming Draft Lesson Plan 	**Suggestion**: 2 teaching hours • Quickly present the course's educational goals. • Discuss with the class the answers to the 3 questions from the first presentation. • Watch the video. • Discuss the video for a while. Ask students what they liked in the video and if they have any questions. • Present the slides of the 2nd presentation, one by one. Ask the students for help in explaining the concepts they have understood from the video. • Watch the 2nd Video. • Discuss it for a while. • Allow students to use slides and install Python on their own computer. • Give students 10 minutes to solve the Crossword. • Quickly discuss the Crossword's solutions. • As an epilogue, discuss the teaching objectives and if they were achieved
 	Leave the quiz for the next lesson. You may give your students the opportunity to complete it at home via their accounts in order to earn points. You will see that the "winning points" process is extremely important to students and they will compete for it. Make sure this is done in an environment with mutual respect and acceptance. You should talk about this in order to avoid very competitive conditions that can harm the classroom.
	**DO NOT FORGET!** Pause the video whenever you feel it is necessary. It is important to clarify any difficult concept at the time it is being talked about. It's good practice to stop a 5 minute video 1 or 2 times to repeat or explain important topics, but also to ensure the increased attention of students. **Crossword**.

	**REMEMBER**. Make sure students enter the answers in capital letters *Answers:* • The main electronic component of the computer is the TRANSISTOR • The person who writes the instructions to the computer is called DEVELOPER • Programs are queues of computer instructions that seek to solve a PROBLEM • The developer, in order to solve this problem, must first think about the steps he or she will take to find the solution. The analytical and clear formulation of these steps is called an ALGORITHM • In Python, which is the function to output content to the console? PRINT • The only language that the computer understand is the…….language. MACHINE
Module 2 – IDLE Programs and Variables Draft Lesson Plan 	**Suggestion**: 2 teaching hours • Start with the quiz of the previous lesson. Do not spend a lot of time here. • Quickly present the course's educational goals. • Watch the video. • Discuss the video for a while. Ask students what they liked in the video and if they have any questions. • Watch the video for the second time. Freeze the video as many times as needed and give students time to follow Stevie's steps. • Solve the first exercise from the "Exercises" module. • Watch the 2nd video. • Discuss the video for a while. Ask students what they liked in the video and if they have any questions. • Present the slides of the 2nd presentation, one by one. Ask the students to explain the concepts they have understood from the video. • Give students time to solve the Hung Game. • As an epilogue, discuss the teaching objectives and if they were achieved.

   	Leave the quiz for the next lesson. You may give your students the opportunity to complete it at home through their accounts to earn points. You will see that the "winning points" process is extremely important to students and they will compete for it. Make sure this is done in an environment with mutual respect and acceptance. You should talk about this in order to avoid very competitive conditions that can harm the classroom. **DO NOT FORGET!** Pause the video whenever you feel it is necessary. It is important to clarify any difficult concept at the time it is being talked about. It's good practice to stop a 5 minute video 1 or 2 times to repeat or explain important topics, but also to ensure the increased attention of students. **Hung Game** **REMEMBER**. Make sure students enter the answers in capital letters. *Answers:* A name that refers to a value is a VARIABLE **Exercise Solutions:** *Exercise 2.* Download from here: https://www.dropbox.com/s/cw5g11arwomh9nh/lesson2ex2.py?dl=0
Module 3 – Expressions and in-built functions Draft Lesson Plan	**Suggestion**: 2 teaching hours • Start with the quiz of the previous lesson. Do not spend a lot of time on this. • Quickly present the course's educational goals.

- Watch the video.
- Discuss the video for a while. Ask students what they liked in the video and if they have any questions.
- In the 2nd watch of the video, freeze it in several places and give students time to follow Stevie's steps.
- Watch the 2nd video.
- Discuss the video for a while. Ask students what they liked in the video and if they have any questions about it.
- Present the slides of the 2nd presentation, one by one. Ask the students to explain the concepts they have understood from the video.
- Give students time to solve the Crossword.
- Solve the exercises one by one. Assign the exercises that you will not be able to handle in the classroom, to be completed at home.
- As an epilogue, discuss the teaching goals and if they were achieved.

Leave the quiz for the next lesson. You may give your students the opportunity to complete it at home through their accounts to earn points.

You will see that the "winning points" process is extremely important to students and they will compete for it. Make sure this is done in an environment with mutual respect and acceptance. You should talk about this in order to avoid very competitive conditions that can harm the classroom.

DO NOT FORGET! Pause the video whenever you feel it is necessary. It is important to clarify any difficult concept at the time it is being talked about. It's good practice to stop a 5 minute video 1 or 2 times to repeat or explain important topics, but also to ensure the increased attention of students.

Crossword

REMEMBER. Make sure students enter the answers in capital letters.

Answers:

- FUNCTION is a block of organised reusable code
- The symbols of numerical operations are the OPERATORS

	• A combination of variables, operators, and values that represents a single result value, is an EXPRESSION **Exercise Solutions:** Exercise 1 Download from here: https://www.dropbox.com/s/fg1reytrmeay3ne/lesson3ex1.py?dl=0 Exercise 2 Download from here: https://www.dropbox.com/s/esegbhfafbjg1nv/lesson3ex2.py?dl=0 Exercise 3 Download from here: https://www.dropbox.com/s/90rcy9sa3hgy066/lesson3ex3.py?dl=0
Module 4 - Edison Robot - Get Familiar and Set Up Draft Lesson Plan 	*If you do not have the Edison robotic system then you can ignore this Module.* *Caution!* *In this case both you and the students should mark this lesson as being finished by pressing "Mark Complete".*  **Suggestion**: 2 teaching hours • Start with the quiz of the previous lesson. Do not spend a lot of time on this. • Quickly present the course's educational goals. • You can choose to watch the video of the 3rd slide of the original slideshow. • If you choose to watch it, talk about the video for a while. Ask students what they liked in the video and if they have any questions. • Watch the 1st video. • Discuss the video for a while. Ask students what they liked in the video and if they have any questions. • Present the slides of the 2nd presentation, one by one. Ask the students to explain the concepts they have understood from the video.

 	• Watch the 2nd video. • Discuss the video for a while. Ask students what they liked in the video and if they have any questions. • Solve the first Exercise from the Module "Exercises" • Watch the 3rd video. • Discuss the video for a while. Ask students what they liked in the video and if they have any questions. • Finish the slides in the last presentation quickly, they are there so that the students may refer to them on an as needed basis. • Take care of the 2nd Exercise. • Leave the Crossword as an introduction to the next lesson if there is no time left. **DO NOT FORGET!** Pause the video whenever you feel it is necessary. It is important to clarify any difficult concept at the time it is being talked about. It's good practice to stop a 5 minute video 1 or 2 times to repeat or explain important topics, but also to ensure the increased attention of students. **Crossword** REMEMBER. Make sure students enter the answers in capital letters. *Answers:* • SENSORS are used to estimate a robot's condition and environment • OUTPUT Sound speakers, lights, drive system and wheels, IR LEDS and more • PROCESSOR is the heart and the brain of a robot • In all Python programs all instructions that start with the # character are considered as COMMENTS
Module 5 – Python and Turtle Programming Draft Lesson Plan	**Suggestion**: 3 teaching hours • Quickly present the course's educational goals. • Take a look at the short presentation just after presenting the goals

- Watch the video.
- Discuss the video for a while. Ask students what they liked in the video and if they have any questions.
- In the 2nd watch of the video, freeze it in several places and give students time to follow Stevie's steps.
- Show the slides of the 3rd presentation, one by one. Give students the chance to make the connection to Python Turtle. Do not work too much on these concepts. They are advanced concepts and we do not want to waste too much time here.
- The purpose of the last presentation is to use it as a reference point for using the appropriate instructions to solve Exercise 1. So pass through this presentation quickly, but encourage students to return in order to find the appropriate instructions to solve Exercise 1.
- Solve Exercise 1.
- If you have time, start with the game "Who Wants to Be a Millionaire". In this game, you will see that there are several repetitive concepts. It is good for students to work with the exercise at home too.
- As an epilogue, discuss the teaching goals and if they were achieved.

Leave the quiz for the next lesson. You may give your students the opportunity to complete it at home through their accounts to earn points.

You will see that the "winning points" process is extremely important to students and they will compete for it. Make sure this is done in an environment with mutual respect and acceptance. You should talk about this in order to avoid very competitive conditions that can harm the classroom.

DO NOT FORGET! Pause the video whenever you feel it is necessary. It is important to clarify any difficult concept at the time it is being talked about. It's good practice to stop a 5 minute video 1 or 2 times to repeat

or explain important topics, but also to ensure the increased attention of students.

Who wants to be a millionaire game

- OPERATORS. The symbols of numerical operations.
- #. Operator for comments in Python
- EXPRESSION. A combination of variables, operators, and values that represents a single result value.
- Maria= turtle.Turtle(). We create a Turtle object and assign it to the Maria variable.
- OBJECT. An instance of a class.
- CLASS. A family of objects with the same properties and characteristics and behaviour
- LOGICAL ERRORS. Programming errors come mostly from the developer. They can be syntax errors that concern the correct use of the programming language syntax or…… that involve errors in capturing the original algorithm.
- CoyoteLearner S.T.E.M Academy. What is the output produced by print() in the following code? print("CoyoteLearner", "S.T.E.M Academy")
- CoyoteLearner - S.T.E.M Academy. What is the output produced by print() in the following code? print("CoyoteLearner"," - ", "S.T.E.M Academy")
- ALGORITHM. A set of specific steps for solving a problem.
- PROGRAM. A sequence of instructions that specifies to a computer actions and computations to be performed.
- VARIABLE. A name that refers to a value.
- FUNCTION. We call a ..... with its name in our program
- Input FUNCTION. Reads a response from the keyboard and assigns it as a string to the variable on the left of the assignment.
- PROPERTIES. Specific features that an object may have. They are usually expressed with variables.
- METHOD. Implementation of specific behaviours of an object. Functions that apply only to these objects.

Exercise Solutions:

Exercise 1
Download from here:

	https://www.dropbox.com/s/9vxb3ihlvwpstqx/lesson5ex1.py?dl=0

Module 6 – Moving Edison Important Note  Draft Lesson Plan  	*If you do not have the Edison robotic system then you can ignore this Module.* *Caution!* *In this case both you and the students should mark this lesson as being finished by pressing "Mark Complete".*  **Suggestion**: 3 teaching hours *Be aware from the beginning that the exercises and challenges that are in this lesson are particularly popular with students and have strong educational incentives. So be sure to leave at least 1 teaching hour for these activities.* • Start with the quiz of the previous lesson. Do not spend a lot of time on this. • Quickly present the course's educational goals. • Watch the 1st video. Freeze it where necessary and follow Stevie's steps. • Discuss the video for a while. Ask students what they liked in the video and if they have any questions. • The purpose of the last presentation is to serve as a reference point for using the appropriate instructions to solve the exercise "Moving Edison Exercise". So go through this presentation quickly but encourage students to go back to it to find the appropriate instructions to resolve the exercise. • Solve exercise 1. Be sure to print the worksheet beforehand. Incorporate Challenge 1 directly into the exercise. • For challenge 2 you will need black insulating tape that students can place on a table top (white is better) or the floor. Alternatively, blank sheets of A3 size could be used. • As an epilogue, discuss the teaching goals and if they were achieved. **DO NOT FORGET!** Pause the video whenever you feel it is necessary. It is important to clarify any difficult concept at the time it is being talked about. It's good practice to stop a 5 minute video 1 or 2 times to repeat or explain important topics but also to ensure the increased attention of the students. **Notes for Edison Robot**

- Because of the minimal mechanical differences in Edison Robot's various editions, some robots may not turn exactly the degrees we define. Encourage students to try different values (eg +3 or -3) to find the value that suits their Edison perfectly.

Exercise Solutions

Exercise 1

Example code:

```
degreesToTurn = 90

Ed.Drive(Ed.FORWARD, Ed.SPEED_5, 13)

Ed.Drive(Ed.SPIN_LEFT, Ed.SPEED_5, degreesToTurn)

Ed.Drive(Ed.FORWARD, Ed.SPEED_5,3)

Ed.Drive(Ed.SPIN_RIGHT, Ed.SPEED_5, degreesToTurn)

Ed.Drive(Ed.FORWARD, Ed.SPEED_5, 15)
```

MODULE 7 – Lists & Iterations Draft Lesson Plan 	**Suggestion**: 3 teaching hours • Quickly present the course's educational goals. • Watch the video. • Discuss the video for a while. Ask students what they liked in the video and if they have any questions. • Show the slides of the 2nd presentation, one by one. Give students the opportunity to test the instructions, shown on these slides, on their computer (these are the instructions shown in the video). • Watch the 2nd video. • Discuss the video for a while. Ask students what they liked in the video and if they have any questions. • Start presenting the slides of the last slide show. Your students should try the instructions presented in the slideshow (the instructions presented in the video), on their computer. • Solve the 1st exercise from the Module "Exercises". • Take care to do the other exercises in order. The last exercise is a challenge and it is important to discuss it. • As an epilogue, discuss the teaching goals and if they were achieved.
Important Note 	Python is very particular about white spaces (e.g. blank characters) inside code lines. In Python syntax, using the tab key to indent a line of code and using the space key to indent it the same amount are not the same. Students should use the tab key. *In Python, all instructions that are just below each other or better, that have, to their left, exactly the same empty space, are considered a single block of code.* Leave the quiz for the next lesson. You may give your students the opportunity to complete it at home through their accounts to earn points. You will see that the "winning points" process is extremely important to students and they will compete for it. Make sure this is done in an environment with


 
 
	mutual respect and acceptance. You should talk about this in order to avoid very competitive conditions that can harm the classroom.
	
	DO NOT FORGET! Pause the video whenever you feel it is necessary. It is important to clarify any difficult concept at the time it is being talked about. It's good practice to stop a 5 minute video 1 or 2 times to repeat or explain important topics but also to ensure the increased attention of the students.
	
	Exercise Solutions:
	Exercise 1
	Download from here:
	https://www.dropbox.com/s/yqb3kyn268g3zny/lesson7ex1.py?dl=0
	
	Exercise 2
	Download from here:
	https://www.dropbox.com/s/r3gmtpab4xsvcr3/lesson7ex2.py?dl=0
	
	Exercises 3 and 4
	
	Advice:
	Remind your students that the sum of the angles of a triangle is 180 degrees. When the triangle is equilateral, the corners are equal. Let's find the angles (180/3)
	
	Correspondingly the sum of the angles of a hex is 720 degrees.
	
	Challenge
	Solution:
	`degreesToTurn = 1`
	`for x in range(360):`
	`  Ed.Drive(Ed.FORWARD, Ed.SPEED_5, 1)`
	`  Ed.Drive(Ed.SPIN_LEFT, Ed.SPEED_5, degreesToTurn)`

MODULE 8 – Loops and While Repetition	**Suggestion**: 4 teaching hours • Start with the quiz of the previous lesson. Do not spend a lot of time on this. • Quickly present the course's educational goals.

Draft Lesson Plan 	• Watch the video. • Discuss the video for a while. Ask students what they liked in the video and if they have any questions. • In the 2nd watch of the video, freeze it in several places and give students time to follow Stevie's steps. • Show the slides of the 2nd presentation, one by one. Discuss the content with your students. • Watch the 2nd video. • Discuss the video for a while. Ask students what they liked in the video and if they have any questions. • Present the slides of the last slide show. Do not waste time here. Encourage students to use these slides as a benchmark for the solution of the following exercises. • Solve the 1st exercise from the Module "Exercises" along with your students. • Have the remaining exercises done by the students in the classroom. Stay there to visit the computers one by one and offer your help to each one of your students. • Leave 15 minutes for the game "Who Wants to Be a Millionaire". • Assign for home all the exercises and games you did not finish in the classroom. • As an epilogue, discuss the teaching goals and if they were achieved.
Important Note 	*Python is very particular about white spaces (e.g. blank characters) inside code lines. In Python syntax, using the tab key to indent a line of code and using the space key to indent it the same amount are not the same. Students should use the tab key.* *In Python, all instructions that are just below each other or better, that have, to their left, exactly the same empty space, are considered a single block of code.*
	*Leave the quiz for the next lesson. You may give your students the opportunity to complete it at home through their accounts to earn points.* *You will see that the "winning points" process is extremely important to students and they will compete for it. Make sure this is done in an environment with*

	*mutual respect and acceptance. You should talk about this in order to avoid very competitive conditions that can harm the classroom.* ***DO NOT FORGET!*** *Pause the video whenever you feel it is necessary. It is important to clarify any difficult concept at the time it is being talked about. It's good practice to stop a 5 minute video 1 or 2 times to repeat or explain important topics but also to ensure the increased attention of the students.* **Who wants to be a millionaire – Solutions** • *In "for" statement, if it is not mentioned in the "range" function, the number that the repeating variable starts with is automatically considered as 0* • *What does the following code fragment write to the monitor?* *a=8* *result = a + 3 * 4 –a* *print (result)* *Answer: 12* • *What keyword would you use to add an alternative condition to an if statement?* *Answer: elif* • *Which one of the following is a valid Python if statement?* *Answer: if a >= 22:* • *What does the following code fragment write to the monitor?* *if (8 <= 7) :* *print ("Hello World")* *elif (9 == 9):* *print ("Hello Robert")* *else:*

	print ("Hello YouTube") *Answer: Hello Robert* • *What does the following code fragment write to the monitor?* *sum = 14* *if (sum < 20):* *print("Under ")* *else:* *print("Over ")* *print("the limit.")* *Answer: Under the limit* • *What does the following code fragment write to the monitor?* *sum = 94* *if (sum < 20):* *print("Under ")* *print("the limit.")* *else:* *print("Over ")* *print("the limit.")* *Answer: Over the limit* • *What does the following code fragment write to the monitor?* *sum = 7*

	if (sum > 20): *[TAB]print("You win ");* *else:* *[TAB]print("You lose ");* *print("the prize.");* *Answer: You lose the prize* • *Say that **a** variable has the value 19 stored in it, and that an **extra** variable has the value 25 stored in it. Evaluate (to true or false) each of the following expressions:* *a <= extra a < extra a > -25 a >= extra* *Answers: true true true false* • *Evaluate (to true or false) each of the following expressions:* *14 <= 14 14 < 14 -9 > -25 -25 > -9* *Answers: true false false false* • *How many choices are possible when using a simple if statement? Answer: one* • *We encounter the if statement in 3 formats. The simple if, the complex if and the ..... if. Answer: Multiple* • *In the logical expression* *age> 10 and age <18 age* *the final result is true only if BOTH the two expressions are …Answer: True* • *What is the result of the following logical expression?* *12 + 23> 31 + 3 Answer: True* • *The ..... expressions, often called conditions, are numerical representations but include at least one comparative operator. More specifically, they include a greater or lesser or equal symbol, or a combination thereof. Answer: logical*

	What will be printed on the screen after you run the following code: *a = 3* *result = a + 3 * 4 -a* *print (result)* *Answer:12* **Exercise Solutions:** *Exercise* 1. Download from here: https://www.dropbox.com/s/wf3oy9vjtkvl5hh/lesson8ex1.py?dl=0 *Exercise* 2. Download from here: https://www.dropbox.com/s/h3qu5a1ajb49srn/lesson8ex2.py?dl=0 *Exercise* 3. Download from here: https://www.dropbox.com/s/5ftncoizxe1d7qd/lesson8ex3.py?dl=0 *Exercise* 4. Download from here: https://www.dropbox.com/s/27sc8lnmnnygrh5/lesson8ex4.py?dl=0 *Exercise* 5. Download from here: https://www.dropbox.com/s/0nyiy1llgnssnt0/lesson8ex5.py?dl=0
MODULE 9 – Loops and While Repetition Draft Lesson Plan	**Suggestion**: 4 teaching hours • Start with the quiz of the previous lesson. Do not spend a lot of time on this. • Quickly present the course's educational goals. • Watch the video. • Discuss the video for a while. Ask students what they liked in the video and if they have any questions.

<table>
<tr>
<td>  </td>
<td>

In the 2nd watch of the video, freeze it in several places and give students time to follow Stevie's steps.
Present the slides of the 2nd presentation, one by one. Discuss the content with your students.
Solve the 1st exercise from the Module "Exercises" along with your students.
Have the remaining exercises done by students in the classroom. Stay there to visit the computers one by one and offer your help to each one of your students.

The sub-module "Edison and While Iteration" is quite demanding. Spend enough time on this.
</td>
</tr>
<tr>
<td>Important Note
  </td>
<td>

Present the slides of the 2nd presentation, one by one. Discuss the content with your students. Try the activities one by one with your robots.
Solve the exercises that are embedded in the presentation.
As an epilogue, discuss the teaching goals and if they were achieved.

</td>
</tr>
<tr>
<td>Important Note
  </td>
<td>Python is very particular about white spaces (e.g. blank characters) inside code lines. In Python syntax, using the tab key to indent a line of code and using the space key to indent it the same amount are not the same. Students should use the tab key.

In Python, all instructions that are just below each other or better, that have, to their left, exactly the same empty space, are considered a single block of code.</td>
</tr>
<tr>
<td> 
  </td>
<td>Leave the quiz for the next lesson. You may give your students the opportunity to complete it at home through their accounts to earn points.

You will see that the "winning points" process is extremely important to students and they will compete for it. Make sure this is done in an environment with mutual respect and acceptance. You should talk about this in order to avoid very competitive conditions that can harm the classroom.</td>
</tr>
</table>

DO NOT FORGET! *Pause the video whenever you feel it is necessary. It is important to clarify any difficult concept at the time it is being talked about. It's good practice to stop a 5 minute video 1 or 2 times to repeat or explain important topics but also to ensure the increased attention of the students.*

Notes for sub-module «Edison & While Iteration»

Remind students of the importance of clearing the stored data when you use a Read function in a loop.

Solution Exercise Slide 5

```
13  Ed.ReadKeypad()
14
15  while Ed.ReadKeypad() == Ed.KEYPAD_NONE:
16      pass
17  Ed.Drive(Ed.FORWARD, Ed.SPEED_6, 8)
18
19  while Ed.ReadKeypad() == Ed.KEYPAD_NONE:
20      pass
21  Ed.Drive(Ed.Ed.BACKWARD, Ed.SPEED_6, 8)
```

Solution Exercise Activity Slide 10 task 1

```
14  while True:
15      Ed.ReadClapSensor()
16      if Ed.ReadClapSensor() == Ed.CLAP_DETECTED:
17          Ed.Drive(Ed.FORWARD, Ed.SPEED_6, 8)
18
```

Solution Exercise Activity Slide 10 task 2

```
14  while True:
15      Ed.Drive(Ed.FORWARD, Ed.SPEED_8, 10)
16      Ed.TimeWait(350, Ed.TIME_MILLISECONDS)
17      Ed.ReadClapSensor()
18      while True:
19          if Ed.ReadClapSensor() == Ed.CLAP_DETECTED:
20              Ed.Drive(Ed.BACKWARD, Ed.SPEED_6, 8)
21              break
```

Solution Exercise 1 Slide 19

```
Ed.ObstacleDetectionBeam(Ed.ON)
Ed.ReadObstacleDetection()
Ed.Drive(Ed.FORWARD, Ed.SPEED_5, Ed.DISTANCE_UNLIMITED)

while Ed.ReadObstacleDetection() != Ed.OBSTACLE_AHEAD:
    pass
Ed.Drive(Ed.SPIN_RIGHT, Ed.SPEED_5, 180)
Ed.Drive(Ed.FORWARD, Ed.SPEED_5, 10)
```

	Solution Exercise 2 Slide 19
	
	`Ed.ObstacleDetectionBeam(Ed.ON)`
	`Ed.ReadObstacleDetection()`
	`Ed.Drive(Ed.FORWARD, Ed.SPEED_5, Ed.DISTANCE_UNLIMITED)`
	
	`while True:`
	`  if Ed.ReadObstacleDetection() == Ed.OBSTACLE_AHEAD:`
	`  Ed.PlayBeep()`

Important Note 	To ensure the best results when using obstacle detection, have students calibrate their Edison using activity sheet 7.1 found here: https://coyotelearner.co/calibrate-obstacle/
Important Note 	Print it before starting this activity. This action is particularly important if the obstacle detection is very sensitive or not sensitive enough. Let's remind students that in order for Edison to be able to locate an obstacle, the obstacles must be opaque but not too dark (e.g. not black) and at least as tall as Edison.
Important Note 	The activity you will find in Slide 20 can be used as a trigger for a discussion about autonomous robotics, artificial intelligence and real-world applications such as robotic cars without a driver, but which use sensors to avoid collisions with people, buildings and other cars. **Exercise Solutions:** *Exercise* 3. Download from here: https://www.dropbox.com/s/rttfovulhcolm74/lesson9ex3.py?dl=0 *Exercise* 4. Download from here: https://www.dropbox.com/s/3ghbibms699viun/lesson9ex4.py?dl=0
MODULE 10 – Functions & Flow Charts	**Suggestion**: 4 teaching hours

Draft Lesson Plan 	• Start with the quiz of the previous lesson. Do not spend a lot of time on this. • Quickly present the course's educational goals. • Watch the video. • Discuss the video for a while. Ask students what they liked in the video and if they have any questions. • In the 2nd watch of the video, freeze it in several places and give students time to follow Stevie's steps. • Present the slides of the 2nd presentation, one by one. Discuss the content with your students. • Solve the Crossword. Use teamwork in the classroom. Run the lines of code that exist in the questions to see their result. • Assign the 1st exercise to the students in the classroom. Give separate help to each student as required.
Important Note 	The sub-module "Edison Functions and Event Driven Programming" is quite demanding. Spend enough time on this. • Present the slides in the classroom, one by one. Discuss the content with your students. Try the activities one by one with your robots. • Solve the exercises that are embedded in the presentation. • Solve the Quiz. • As an epilogue, discuss the teaching goals and if they were achieved.
	*You will see that the "winning points" process is extremely important to students and they will compete for it. Make sure this is done in an environment with mutual respect and acceptance. You should talk about this in order to avoid very competitive conditions that can harm the classroom.*
	***DO NOT FORGET!*** *Pause the video whenever you feel it is necessary. It is important to clarify any difficult concept at the time it is being talked about. It's good practice to stop a 5 minute video 1 or 2 times to repeat or explain important topics but also to ensure the increased attention of the students.* **Crossword**.

REMEMBER. Make sure students enter the answers in capital letters

- We can call .... as many times as we want with different arguments each time. Answer: FUNCTIONS
- When calling a function, we put variables or values in place of the parameters to be used to make the calculations. These values are called ..... Answer: ARGUMENTS
- When calling a function, an argument can be either a value or a ..... Answer: VARIABLE
- They enclose the input parameters or arguments of a function. Answer: PARENTHESES
- A keyword that marks the beginning of function block. Answer: Def
- In the following code snippet, number1, number2 and number3 are:

```
def max(number1, number2, number3):
    if number1>number2 and number1>number3:
        print('maximum is: ', number1)
    elif number2>number1 and number2>number3:
        print('maximum is: ', number2)
    else:
        print('maximum is: ', number3)

max(a,b,c)
```

Answer: PARAMETERS

- What is the result of the f() function in the following code snippet?

```
num = 4
def f():
    num = 3
print(num)

f()
print(num)
```

Answer: NOTHING

- What is the output of the following code snippet? (Please use words in your answer, example: FIVESIX)

```
num = 4
def f():
    num = 3
print(num)

f()
print(num)
```

Answer: FOURFOUR

- What is the output of the following code snippet? (Please use words in your answer, example: FIVEFIVE for 55)

```
num = 4
def f():
    num = 3
    print(num)

f()
print(num)
```

Answer: THREEFOUR

Exercise Solutions:

Exercise 1.
Download from here:
https://www.dropbox.com/s/gd86duq6un2z2vi/lesson10ex1.py?dl=0

Notes for sub-module «Edison Functions and event…»

Solution: Exercise 1 Slide 6

```
while True:
    if Ed.ReadClapSensor() == Ed.CLAP_DETECTED:
        dosomething()
        dosomething()
        break

def dosomething():
    Ed.LeftLed(Ed.ON)
    Ed.Drive(Ed.SPIN_RIGHT, Ed.SPEED_5, 90)
    Ed.RightLed(Ed.ON)
    Ed.PlayBeep()
    Ed.Drive(Ed.SPIN_LEFT, Ed.SPEED_5, 90)
    Ed.LeftLed(Ed.OFF)
    Ed.RightLed(Ed.OFF)
```

Solution: Exercise 2 Slide 6

```
squere(5)
squere(10)
squere(20)

def squere (squareside):
    Ed.Drive(Ed.FORWARD, Ed.SPEED_6, squareside)
    Ed.Drive(Ed.SPIN_RIGHT, Ed.SPEED_6, 90)
    Ed.Drive(Ed.FORWARD, Ed.SPEED_6, squareside)
    Ed.Drive(Ed.SPIN_RIGHT, Ed.SPEED_6, 90)
    Ed.Drive(Ed.FORWARD, Ed.SPEED_6, squareside)
    Ed.Drive(Ed.SPIN_RIGHT, Ed.SPEED_6, 90)
    Ed.Drive(Ed.FORWARD, Ed.SPEED_6, squareside)
    Ed.Drive(Ed.SPIN_RIGHT, Ed.SPEED_6, 90)
```

MODULE 12
Parking Edison

Draft Lesson Plan

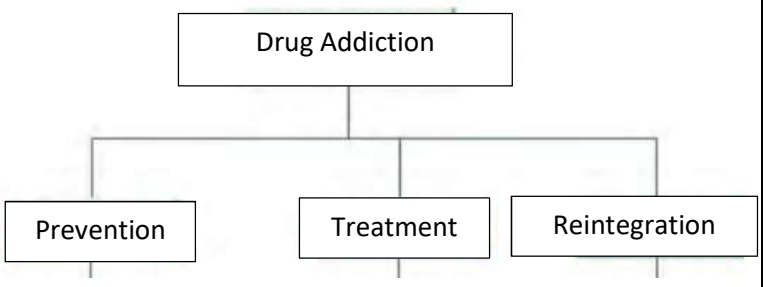
Suggestion: 5 teaching hours

Be sure to print the track (the Parking lot). Prefer A3 printing on thick matte paper.

First teaching hour

- Present briefly the educational goals of the course.
- Discuss in the classroom the slides under "Troubleshooting Methodology" section.
- Ask students to think about general problems. Encourage them to analyze them in subproblems. You may use as an example the problem of drug addiction
  (Dedicate here about 15 minutes)

The problem of drug addiction can be broken down into three sub-problems. Those 3 sub-problems can be divided into more. Share with your class the following diagrammatic representation of this problem:



	It is obvious that the **diagrammatic representation** can lead to a better understanding of the problem and its structure. Be sure to discuss this with your students. Please note that the sub-problems may and should be divided into simpler parts. Do not waste any more time here though. • Then, by using the slides of the section "The Problem of Automatic Parking" discuss with the class the problem you have to deal with, in this module.
Second teaching hour	• Discuss briefly again the "auto-parking Edison" problem you have to deal with through the slides and video of the corresponding subsection. It is important for students to understand the problem. • Proceed to the diagrammatic representation of the problem. Insist on slides 1 and 2 of the 3rd subsection. Make sure that students understand the structure of the problem. • Proceed to the solution of the first sub-problem. Ask your students for the flow chart before introducing it to the classroom with the corresponding slide. • Program Edison and try the solutions inside the track.
Third teaching hour	• Proceed to the solution of sub-problem 2. • Ask your students for the flow chart before presenting to them the corresponding slide. • Solve the problems 2.1 and 2.2 and then the sub-problem 2. • Program Edison and try the code on the printed track. • Simultaneously program some Edisons to solve the subproblem 1 and some other to solve the subproblem 2. Try the codes on the track. • Compare solutions. Think about how to combine those solutions later.
Fourth teaching hour	• Proceed to the solution of sub-problem 3. • Ask your students for the flow chart before presenting to them the corresponding slide. • Program Edison and try the code on the printed track.

<table>
<tr>
<td>Fifth teaching hour</td>
<td>

Simultaneously program some Edisons to solve the subproblem 1, some other to solve the subproblem 2 and some more to solve the third sub-problem.
Compare solutions. Think about how to combine those solutions later.

Proceed to the composition of the overall solution.
Program Edison and try the code on the track.
Discuss the code lines that were not used in the subproblems solutions. For the discussion, use the slides 4, 5 and 6 of the "Problem Solution" section.
Solve the exercise only if there is enough time for it. Indicatively, a simple solution is to include the solution of the third sub-problem in a function.

<pre>
def leaveparking():
    #start driving forwards
    Ed.Drive(Ed.FORWARD, Ed.SPEED_1, Ed.DISTANCE_UNLIMITED)
    #while Edison is on a white surface continue forwards
    while Ed.ReadLineState()==Ed.LINE_ON_WHITE:
        pass
    #when Edison is not on a white surface stop, turn left
    Ed.Drive(Ed.STOP, Ed.SPEED_10, 0)
    Ed.Drive(Ed.SPIN_LEFT, Ed.SPIN_LEFT, 90)
    Ed.Drive(Ed.FORWARD, Ed.SPEED_1, Ed.DISTANCE_UNLIMITED)
    #while Edison is on a white surface continue forwards
    while Ed.ReadLineState()==Ed.LINE_ON_WHITE:
        pass
    #when Edison is not on a white surface stop, Edison has
    Ed.Drive(Ed.STOP, Ed.SPEED_10, 0)
</pre>

As an epilogue, discuss the teaching goals and if they were achieved.

</td>
</tr>
</table>