**Programming in Python with the Initio Robot: Part 1 Answer Booklet**

**Chapter 1**

**Potential Problems:** Assuming the Initio robot has been correctly constructed and is not broken then there should be no significant problems locating ports and connecting up the robot.

**Chapter 2**

**Sample Answers:** The questions here are all descriptive and should be easy to answer provided the student is watching the robot.

**Potential Problems:**

* Confusion may be caused by students failing to press Return after entering a command.
* Because the robots are tethered to monitor, keyboard and mouse they may not move very steadily.

**Chapter 3**

**Sample Answers:** The questions here are all descriptive and should be easy to answer once the student has correctly understood what to type.

**Potential Problems:**

* Confusion may be caused by students failing to press Return after entering a command.
* Confusion may be caused by the use of argument for *speed* etc. in the commands. This may need to be explained to students.
* Weaker students may not like the open nature of the final question and perhaps will need direction to try out a specific set of commands and see what they do.

**Chapter 4**

**Sample Answers:** The questions here are all descriptive again and the difficulties may lie in understanding how to change the values returned by the various sensors.

**Potential Problems:**

* Objects representing obstacles need to be in front of the actual sensors. So, for instance, the ultrasonic sensor will not detect a small object on the ground in front of the robot. The object needs to be tall enough to reach the height of the sensor.
* See through objects (such as water bottles) may not be picked up by the Infrared sensors which are using light as their detection mechanism.
* The angle at which the sound from the ultrasonic sensors and the light from the infra-red sensors hits an object can cause errors (at some angles the sound/light will bounce away from the robot rather than back towards the robot). This is explored in Ex4-Initio-SensorAngles which suggests experimenting to discover this angle. **IMPORTANT:** This limitation isn’t present in the in the simulator which may cause confusion for some students if they are moving between the two.

**Chapter 5**

These exercises take the form of experiments to understand the limitations of the sensors. There may be some logistical difficulties – particularly measuring distances.

For the second exercise (with the infra-red) sensor it is probably better to have an upright flat surface (e.g., a book) placed in front of the robot that can be rotated through an angle, rather than trying to rotate the robot in relation to some object.

**Chapter 6**

**Troubleshooting:** The biggest cause of bugs in the exercises in WS5 are likely to be spelling errors (and possibly spacing errors if the students indent anything for some reason). If they do get errors, they should be encouraged to look at the line number indicated in the error message and check for spelling.

The IDLE IDE is good at highlighting syntax errors and the like. If the program won’t even run, then students should be encouraged to look at the parts of the code highlighted in the IDE.

**Sample Answer 1:** The robot pans its ultrasonic sensor 45 degrees. Bright students may realise that although the program contains initio.forward(10) they don’t see this execute because it is interrupted by initio.stop() soon after.

**Sample Answer 2:**

import robohat as initio

initio.init()

initio.forward(10)

initio.setServo(0, -45)

initio.stop()

initio.setServo(0, 0)

**Potential Problems:** If the ultrasonic servo starts at 0 in the new program (or -45 in the first program) then the program will appear to do nothing at all. Until WS6, there are no commands available to delay the execution of the next command in an Initio program. There is obviously quite a lot of scope for confusion here that may need explaining.

**Chapter 7**

**Troubleshooting:** Students can still type commands into the IDLE window (e.g. initio.stop()) even after running a program from a file.

**Sample Answer 1:** The robot moves forward for 10 seconds and then stops.

**Sample Answer 2:** Changed the input for time.sleep(10) to time.sleep(20)

**Sample Answer 3:**  Note that the wait at the start allows time to unplug the keyboard, mouse and monitor from the robot.

import robohat as initio, time

initio.init()

time.sleep(30)

initio.forward(10)

time.sleep(10)

initio.spinRight(10)

time.sleep(10)

initio.stop()

**Chapter 8**

**Sample Answer 1:**

import robohat as initio, time

initio.init()

initio.spinRight(10)

time.sleep(2)

print(initio.getDistance())

initio.stop()

**Sample Answer 2:**

import robohat as initio, time

initio.init()

initio.setServo(0, 20)

time.sleep(5)

initio.setServo(0, -20)

time.sleep(5)

initio.setServo(0, 10)

**Sample Answer 3:** Note that the answers may vary depending upon the speed the robot turns.

This answer includes a wait at the start in order to allow the robot to be unplugged and pauses each time readings are taken to help the programmer tell if a quarter circle has been turned. Spin speed and timings will depend on battery power.

import robohat as initio, time

initio.init()

time.sleep(30)

print(initio.irLeft())

print(initio.irRight())

initio.spinLeft(60)

time.sleep(2)

initio.stop()

time.sleep(5)

print(initio.irLeft())

print(initio.irRight())

initio.spinLeft(60)

time.sleep(2)

initio.stop()

time.sleep(5)

print(initio.irLeft())

print(initio.irRight())

initio.spinLeft(60)

time.sleep(2)

initio.stop()

time.sleep(5)

print(initio.irLeft())

print(initio.irRight())

initio.spinLeft(60)

time.sleep(2)

initio.stop()

**Sample Answer 4:** Note speeds may vary!

import robohat as initio, time

initio.init()

time.sleep(20)

initio.forward(10)

time.sleep(10)

initio.forward(20)

time.sleep(10)

initio.forward(30)

time.sleep(10)

initio.stop()

**Sample Answer 5:**

import robohat as initio, time

inito.init()

time.sleep(20)

initio.setServo(0, 20)

time.sleep(5)

initio.setServo(0, -20)

time.sleep(5)

initio.setServo(0, 10)

inito.forward(10)

time.sleep(10)

initio.setServo(0, 20)

time.sleep(5)

initio.setServo(0, -20)

time.sleep(5)

initio.setServo(0, 10)

inito.spinLeft(10)

time.sleep(10)

initio.setServo(0, 20)

time.sleep(5)

initio.setServo(0, -20)

time.sleep(5)

initio.setServo(0, 10)

initio.reverse(10)

time.sleep(10)

pi2go.stop()

**Sample Answer 6:** Note answers will vary depending upon the obstacle course created, and the robot speed.

import robohat as initio, time

initio.init()

time.sleep(20)

initio.spinRight(60)

time.sleep(2)

initio.forward(10)

time.sleep(10)

initio.spinLeft(60)

time.sleep(2)

initio.forward(10)

time.sleep(15)

initio.stop()

**Chapter 9**

**Sample Answer 1:** The robot moves forward for 10 seconds or prints the value from the distance sensor and move backwards for 10 seconds (answer depends upon whether the student had something in front of the robot or not)

**Sample Answer 2:** To test the robot was working you need to test its behaviour both when there is an obstacle (it should move backward) and when there isn’t (it should move forward).

**Sample Answer 3:**

import robohat as initio, time

initio.init()

if (initio.irLeft()):

 initio.spinRight(10)

elif (initio.irRight()):

 initio.spinLeft(10)

time.sleep(10)

initio.stop()

Three cases are needed to test this (something on the left, something on the right and nothing on either side).

The description of the exercise doesn’t say what to do if there is an obstacle on both sides of the robot.

**Sample Answer 4:**

import robohat as initio, time

initio.init()

if (initio.getDistance() < 30):

 initio.reverse(10)

 time.sleep(10)

 initio.spinRight(10)

 time.sleep(10)

 initio.stop()

else:

 initio.forward(10)

**Chapter 10**

**Sample Answer 1:** If there is nothing in front of the robot it does nothing. Otherwise it reverses and keeps printing out “reversing” until it is far enough away from the obstacle to stop.

**Sample Answer 2:** You would need to test it when there is an obstacle in front of it and when there isn’t an obstacle in front of it.

**Chapter 11**

**Sample Answer 1:** The program prints “Waiting” until something comes closer than 5cm. Then it “nods its head”.

**Sample Answer 2:** Notice that three while loops are needed - wait for the switch to be switched on, wait for the switch to be switched off and then wait for the switch to be switched on again. Students may need some help thinking through this.

import robohat as initio, time

initio.init()

while (initio.getDistance() > 5):

 print(“Waiting”)

initio.setServo(1, 20)

while not (initio.getDistance() > 5):

 print(“Waiting for Obstacle to Move”)

while (initio.getDistance() > 5):

 print(“Still Waiting”)

initio.setServo(1, 0)

**Sample Answer 3:** not (initio.getDistance() > 5)

**Program:**

import robohat as initio, time

initio.init()

if not (initio.getDistance() > 5):

 initio.forward(10)

while (initio.getDistance() > 5):

 print(“Waiting for Obstacle”)

initio.stop()

**Sample Answer 4:** not (initio.irLeft() or initio.irRight())

or alternatively (not (initio.irLeft()) and not (initio.irRight())

**Program:**

import robohat as initio, time

initio.init()

if (not (initio.irLeft() or initio.irRight())):

 initio.forward(10)

while (not (initio.irLeft() or initio.irRight())):

 print(“Still Advancing”)

initio.stop()

**Chapter 12**

**Sample Answer 1:** The program prints “Waiting” until the switch is pressed. Then it “waves its head”.

**Sample Answer 2:** Notice that three while loops are needed - wait for the distance to be less than 50, wait for the distance to be greater than 50 and then wait for the distance to be less than 50 again. Students may need some help thinking through this.

import robohat as initio, time

initio.init()

while (initio.getDistance() > 50):

 print(“Waiting”)

initio.setServo(1, 20)

while not (initio.getDistance() > 50):

 print(“Waiting for Obstacle to Move”)

while (initio.getDistance() > 50):

 print(“Still Waiting”)

initio.setServo(1, 0)

**Sample Answer 3:** not (initio.getDistance() > 5)

**Program:**

import robohat as initio, time

initio.init()

if not (initio.getDistance() > 50):

 initio.forward(10)

while (initio.getDistance() > 50):

 print(“Waiting for Obstacle”)

initio.stop()

**Sample Answer 4:** not (initio.irLeft() or initio.irRight())

or alternatively (not (initio.irLeft()) and not (initio.irRight())

**Program:**

import robohat as initio, time

initio.init()

if (not (initio.irLeft() or initio.irRight())):

 initio.forward(10)

while (not (initio.irLeft() or initio.irRight())):

 print(“Still Advancing”)

initio.stop()

**Chapter 13**

**Sample Answer 1:** While neither infra-red sensor detects an obstacle, it moves forward and then sleeps for 10 seconds. If either sensor detects an obstacle, then it stops.

**Sample Answer 2:** It does nothing until the left infra-red sensor detects something. Then it moves forward until the right infra-red sensor detects something. Then it stops.

**Sample Answer Exercise 1:**

import robohat as initio

initio.init()

initio.forward(10)

while True:

 if (initio.irLeft()):

 break

initio.stop()

**Sample Answer Exercise 2:**

import robohat as initio

import time

initio.init()

while True:

 if (initio.irLeft() and initio.irRight()):

 break

 if (initio.irLeft()):

 initio.spinRight(10)

 elif (initio.irRight()):

 initio.spinLeft(10)

 else:

 initio.forward(10)

 continue

 print("Spinning to find a clear route")

**Chapter 14**

**Sample Answer Exercise 1:**

import robohat as initio

initio.init()

if (initio.irLeft()):

 initio.setServo(1, 20)

if (initio.irRight()):

 initio.setServo(1, -20)

**Sample Answer Exercise 2:**

import robohat as initio

import time

initio.init()

while not (initio.irLeft()):

 continue

time.sleep(2)

while not (initio.irRight()):

 print(initio.getDistance())

**Sample Answer Exercise 3:**

import robohat as initio

initio.init()

while True:

 if (initio.irLeft()):

 initio.setServo(1, 20)

 if (initio.irRight()):

 initio.setServo(1, -20)

**Sample Answer Exercise 4:**

import robohat as initio

initio.init()

while (initio.irLeft() and initio.irRight()):

 initio.reverse(10)

initio.stop()

**Sample Answer Exercise 5:**

import robohat as initio

initio.init()

while not (initio.irLeft() or initio.irRight() or initio.getDistance() < 50):

 initio.forward(10)

initio.stop()

**Sample Answer Exercise 6:**

import robohat as initio

initio.init()

while True:

 while not (initio.irLeft() or initio.irRight() or initio.getDistance() < 50):

 initio.forward(10)

 while (initio.irLeft() or initio.irRight() or initio.getDistance() < 50):

 initio.reverse(10)

**Chapter 15**

**Sample Answer 1:**

Sensor: getDistance(), Motor: forward(10) and spinLeft(10) or spinRight(10)(speed values may vary)

**The Program:**

import robohat as initio, time

initio.init()

time.sleep(30)

while True:

 while (initio.getDistance() > 50):

 initio.forward(10)

 while not (initio.getDistance() > 50):

 initio.spinLeft(30)

**Program Modified to Stop Cleanly**

import robohat as initio, time

initio.init()

while (initio.getDistance() > 5):

 continue

while not (initio.getDistance() > 5):

 continue

while (initio.getDistance() > 5):

 while (initio.getDistance() > 50):

 initio.forward(10)

 while not (initio.getDistance() > 50):

 initio.spinLeft(30)

initio.stop()

**Exercise 2:**

import robohat as initio, time

initio.init()

while (initio.getDistance() > 5):

 continue

while not (initio.getDistance() > 5):

 continue

while (initio.getDistance() > 5):

 while (initio.irLeftLine()):

 initio.spinLeft(10)

 while (not initio.irLeftLine() and not initio.irRightLine()):

 initio.forward(30)

 while (initio.irRightLine()):

 initio.spinRight(30)

initio.stop()



 University of Liverpool, 2019

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).