Name: ____ **Date:** __________________

Computer Science Level 1 - Python Course

Unit 8 Exam

1. What creates the magnetic field for the Electromagnet?

a. An electric current.

b. An electrical outlet.

c. A button on the VR Robot.

d. An additional magnet.

---

2. What objects on the VR Playground can the Electromagnet on the VR Robot pick up?

a. Disks

b. Buildings

c. Balls

d. Pen drawings

---

3. What is one use of the Electromagnet on the VR Robot?

a. To identify the color of disks.

b. To solve the Wall Maze.

c. To pick up and put down disks with metal cores.

d. To find metal in the walls of Playgrounds.

---

4. What is the best description of the *energize* command?

a. Sets the Electromagnet to two different modes: Left or Right.

b. Energizes the Electromagnet to press both Bumper Sensors.

c. Energizes the Electromagnet to attract all metal objects in a Playground.

d. Sets the Electromagnet to two different modes: ‘BOOST’ or ‘DROP’.

---

5. **What happens when the *energize* command is set to ‘BOOST’ while near a disk?**
   a. The Electromagnet attracts the disk.
   b. The Electromagnet drops the disk.
   c. The Dashboard reports the disk being picked up.
   d. Nothing happens.

---

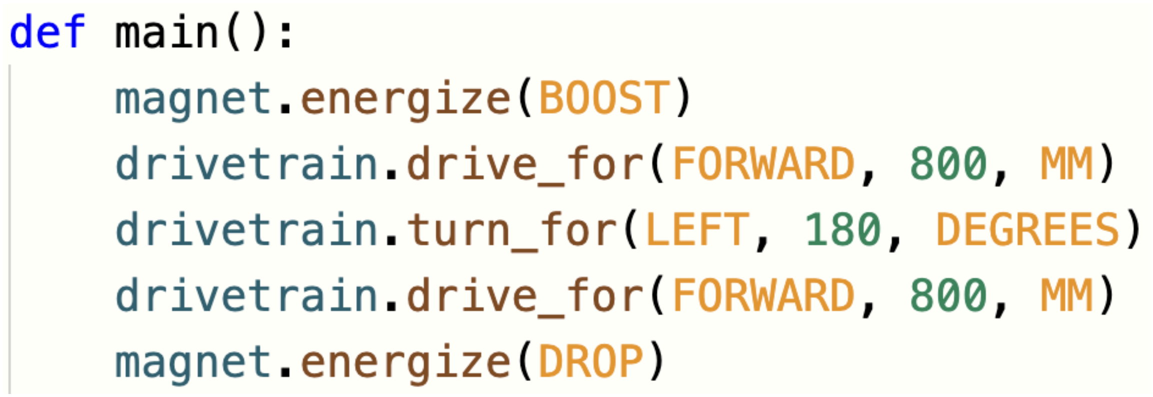
6. **What happens when the *energize* command is set to ‘DROP’ while the Electromagnet is holding a disk?**
   a. Nothing happens.
   b. The Dashboard reports the disk being released.
   c. The Electromagnet picks up the disk.
   d. The Electromagnet releases the disk.

---

7. **True or False: The *energize* command returns a Boolean value that can be used with Drivetrain commands to pick up disks.**
   a. True
   b. False

---

8. **Which of the following best describes the use of comments in a project?**
   a. To turn on the Electromagnet in a project.
   b. To plan and organize the behaviors needed to complete the goal of the project.
   c. To make a project more disorganized and less efficient.
   d. To report the value of a sensor on the VR Robot.



```
def main():
    magnet.energize(BOOST)
    drivetrain.drive_for(FORWARD, 800, MM)
    drivetrain.turn_for(LEFT, 180, DEGREES)
    drivetrain.drive_for(FORWARD, 800, MM)
    magnet.energize(DROP)
```

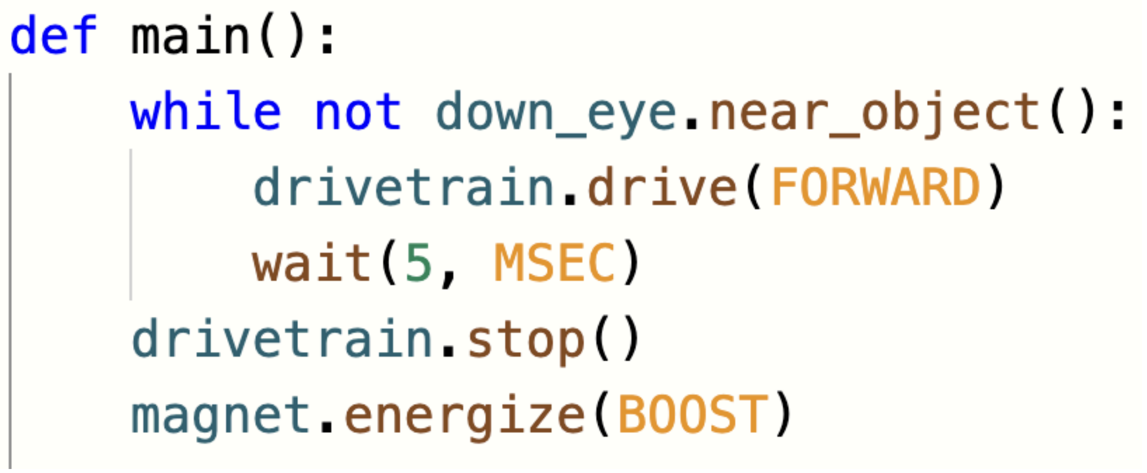
9. Which of the following best describes what the VR Robot will do in this project?

a. The VR Robot will drive forward 800 millimeters (mm), turn left 180 degrees, drive forward 800 millimeters (mm) and drop any disks that were picked up.

b. The VR Robot will energize the magnet, drive forward 800 millimeters (mm), turn left 180 degrees, drive forward 800 millimeters (mm) and drop any disks that were picked up.

c. The VR Robot will energize the magnet, drive forward 800 millimeters (mm), and turn left 180 degrees.

d. The VR Robot will energize the magnet, drive forward 200 millimeters (mm), turn left 90 degrees, drive forward 200 millimeters (mm) and drop any disks that were picked up.



```
def main():
    while not down_eye.near_object():
        drivetrain.drive(FORWARD)
        wait(5, MSEC)
    drivetrain.stop()
    magnet.energize(BOOST)
```

10. In this project, what will happen if the Down Eye Sensor does NOT detect an object?

a. The VR Robot will check the condition of the ***front_eye.detects*** command again.

b. The VR Robot will drive forward.

c. The VR Robot will stop driving.

d. The VR Robot will stop driving, then set the Electromagnet to ‘BOOST.’

11. Which VR Robot sensor determines the color of an object when close enough?

a. Gyro Sensor.
b. Distance Sensor.
c. Eye Sensor.
d. Location Sensor.

---

12. Which sensor does the VR Robot use with the *turn_to_heading* command?

a. Distance Sensor.
b. Gyro Sensor.
c. Location Sensor.
d. Bumper Sensor.

---

13. Which direction will the VR Robot turn when using the *turn_to_heading* command?

a. The VR Robot will determine which direction to turn based on the current heading of the Distance Sensor.

b. The VR Robot will always turn counter-clockwise.

c. The VR Robot will always turn clockwise.

d. The VR Robot will determine which direction to turn based on the current heading of the Gyro Sensor. The VR Robot will turn in the direction with the smaller degree difference between the current and given heading.

14. Which of the following describes one reason to use a *turn_to_heading* command versus a *turn* or *turn_for* command?

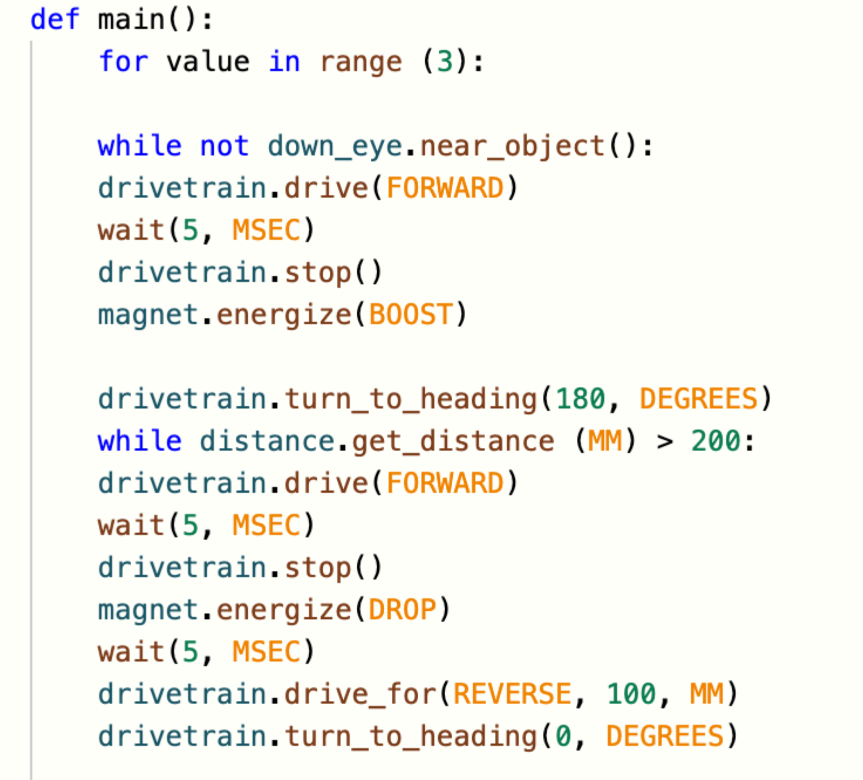
a. ***turn_to_heading*** makes projects repeatable from one run to another because the VR Robot will move to the specified heading regardless of the previous position of the VR Robot.

b. ***turn_to_heading*** will orient the VR Robot to the starting position in any Playground.

c. ***turn_to_heading*** makes projects unique and difficult to repeat from one run to another because the VR Robot will move for a specific number of degrees based on the current position of the VR Robot.

d. ***turn_to_heading*** is based on the readings from the Distance Sensor, so it is more accurate than other turn commands.

15. This project is intended to collect three disks and return them to the goal, but it is not working. What needs to be corrected for this project to run as intended?

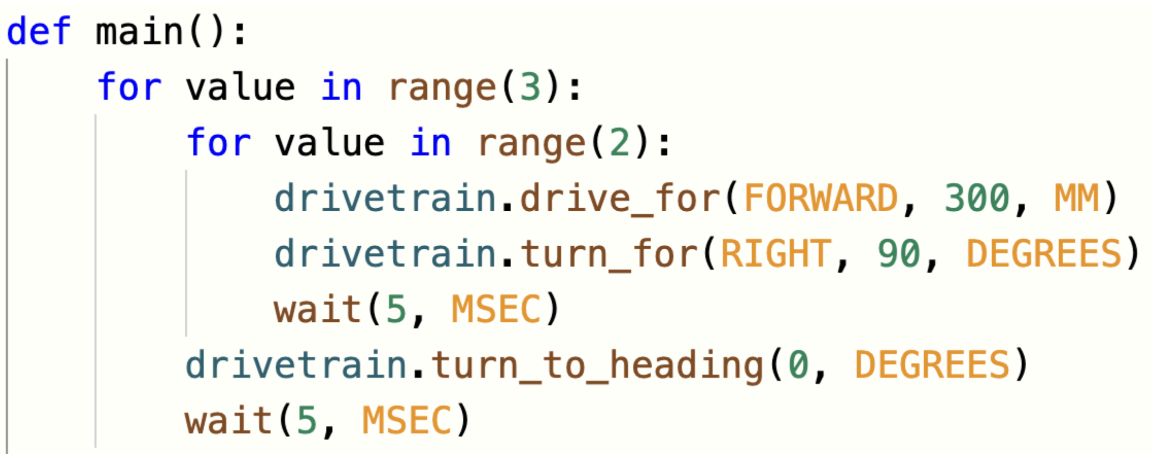


```
def main():
    for value in range (3):

    while not down_eye.near_object():
    drivetrain.drive(FORWARD)
    wait(5, MSEC)
    drivetrain.stop()
    magnet.energize(BOOST)

    drivetrain.turn_to_heading(180, DEGREES)
    while distance.get_distance (MM) > 200:
    drivetrain.drive(FORWARD)
    wait(5, MSEC)
    drivetrain.stop()
    magnet.energize(DROP)
    wait(5, MSEC)
    drivetrain.drive_for(REVERSE, 100, MM)
    drivetrain.turn_to_heading(0, DEGREES)
```

a. The commands within the ***for*** loop and the ***while*** loops need to be properly indented.
b. The ***for*** loop needs to be moved to the bottom of the project.
c. Additional Drivetrain commands need to be added to the project.
d. Only the ***wait*** commands need to be indented further.



```
def main():
    for value in range(3):
        for value in range(2):
            drivetrain.drive_for(FORWARD, 300, MM)
            drivetrain.turn_for(RIGHT, 90, DEGREES)
            wait(5, MSEC)
        drivetrain.turn_to_heading(0, DEGREES)
        wait(5, MSEC)
```

16. Which of the following best describes the project flow in this project?

a. The VR Robot will begin by moving forward 300 millimeters (mm) and will turn right 90 degrees. The VR Robot will then turn to a heading of 0 degrees. The VR Robot will repeat all these steps two additional times for a total of three repeats.

b. The VR Robot will begin by moving forward 300 millimeters (mm) and will turn right 90 degrees. These two commands will be executed two times in a row because of the ***for*** loop. The VR Robot will then turn to a heading of 0 degrees. The VR Robot will repeat all these steps two additional times for a total of three repeats.

c. The VR Robot will begin by moving forward 300 millimeters (mm) and will turn right 90 degrees. The VR Robot will then turn to a heading of 0 degrees. The VR Robot will repeat all these steps one additional time for a total of two repeats.

d. The VR Robot will begin by moving forward 300 millimeters (mm) and will turn right 90 degrees. These two commands will be executed two times in a row because of the ***for*** loop. The VR Robot will then turn to a heading of 0 degrees.

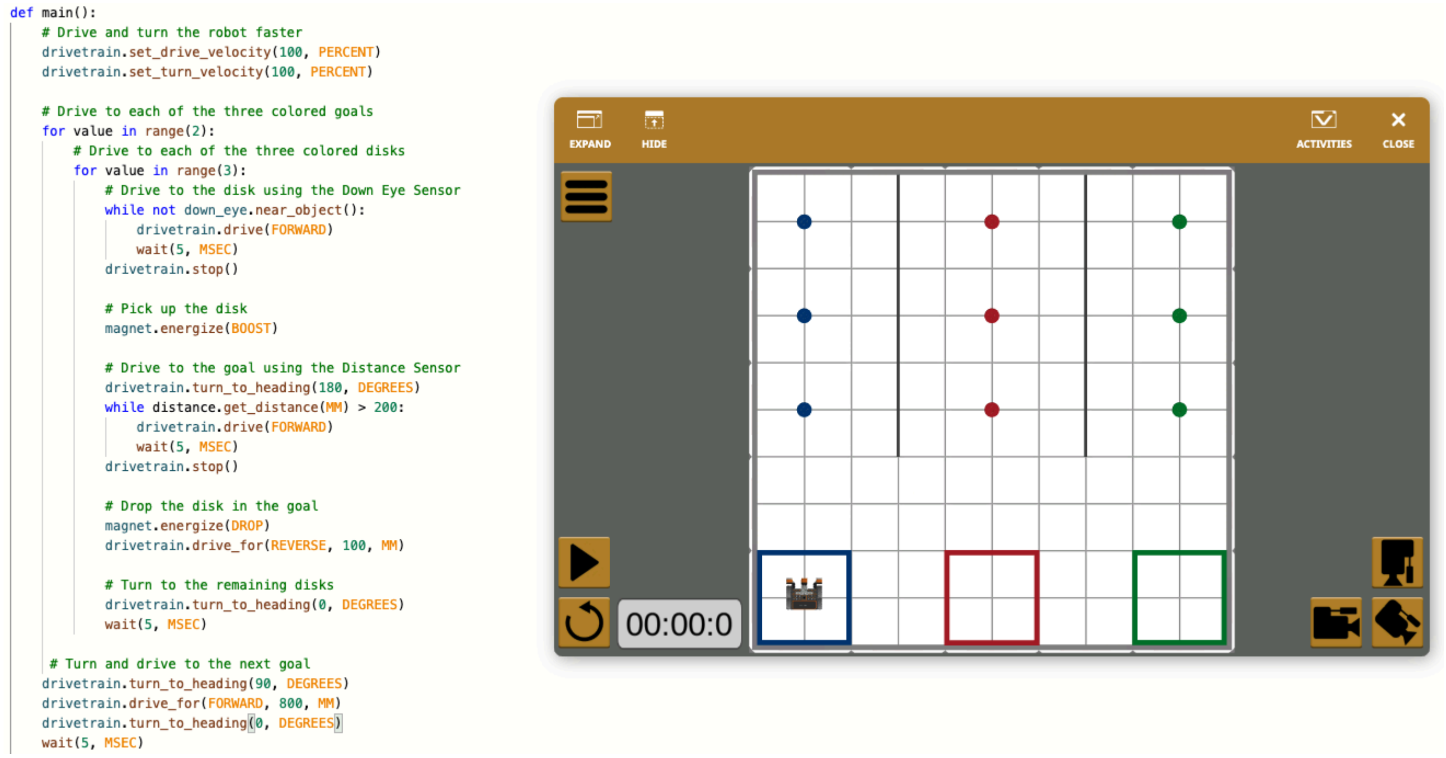
17. What is one reason to use nested loops?

a. To repeat certain actions more than others within the outer ***for*** loop.

b. To sense the distance between the VR Robot and an object.

c. To determine if there is a red object on the Playground.

d. To turn the VR Robot to a specific heading based on the Gyro Sensor.



```
def main():
    # Drive and turn the robot faster
    drivetrain.set_drive_velocity(100, PERCENT)
    drivetrain.set_turn_velocity(100, PERCENT)

    # Drive to each of the three colored goals
    for value in range(2):
        # Drive to each of the three colored disks
        for value in range(3):
            # Drive to the disk using the Down Eye Sensor
            while not down_eye.near_object():
                drivetrain.drive(FORWARD)
                wait(5, MSEC)
            drivetrain.stop()

            # Pick up the disk
            magnet.energize(BOOST)

            # Drive to the goal using the Distance Sensor
            drivetrain.turn_to_heading(180, DEGREES)
            while distance.get_distance(MM) > 200:
                drivetrain.drive(FORWARD)
                wait(5, MSEC)
            drivetrain.stop()

            # Drop the disk in the goal
            magnet.energize(DROP)
            drivetrain.drive_for(REVERSE, 100, MM)

            # Turn to the remaining disks
            drivetrain.turn_to_heading(0, DEGREES)
            wait(5, MSEC)

     # Turn and drive to the next goal
    drivetrain.turn_to_heading(90, DEGREES)
    drivetrain.drive_for(FORWARD, 800, MM)
    drivetrain.turn_to_heading(0, DEGREES)
    wait(5, MSEC)
```

18. This project should solve the Disk Mover Challenge, but it does not work correctly. What is the error?

a. The inner loop is set to repeat three times rather than two.

b. The Electromagnet is never instructed to pick up a disk.

c. The outer loop is set to repeat two times rather than three.

d. The Electromagnet is never instructed to drop a disk.