

Name : _____

Grading TA: _____

- **DEVICES:** If your cell phone, pager, PDA, beeper, iPod, or similar item goes off during the exam, you will lose 10 points on this exam. Turn all such devices off and put them away now. You cannot have them on your desk.
- **ACADEMIC MISCONDUCT:** Academic misconduct will not be tolerated. You are to uphold the honor and integrity bestowed upon you by the Georgia Institute of Technology.
 - Keep your eyes on your own paper.
 - Do your best to prevent anyone else from seeing your work.
 - Do NOT communicate with anyone other than a proctor for ANY reason in ANY language in ANY manner.
 - Do NOT share ANYTHING during the exam. (This includes no sharing of pencils, paper, erasers).
 - Follow directions given by the proctor(s).
 - Stop all writing when told to stop. Failure to stop writing on this exam when told to do so is academic misconduct.
 - Do not use notes, books, calculators, etc during the exam.
- **TIME:** Don't get bogged down by any one question. If you get stuck, move on to the next problem and come back once you have completed all of the other problems. This exam has 7 questions on 10 pages including the title page. Please check to make sure all pages are included. You will have 50 minutes to complete this exam.

INTEGRITY: By taking this exam, you pledge that this is your work and you have neither given nor received inappropriate help during the taking of this exam in compliance with the Academic Honor Code of Georgia Tech. I have also read and understand the requirements outlined above.
Do NOT sign nor take this exam if you do not agree with the honor code.

Signature: _____

Question	Points	Score
1. Vocabulary	9	
2. Dictionaries & Tuples	10	
3. Legal or not?	9	
4. removeFirst	5	
5. Short Answer	11	
6. studentReadLines	10	
7. Run Away	20	
Total:	74	

1. (9 points)

For each of the following vocabulary terms, write a concise 1-2 sentence definition. Be brief, and to the point.

- (a) [3 pts] algorithm

Solution: A finite set of unambiguous instructions performed in a prescribed sequence to achieve a goal.

- (b) [3 pts] boolean expression

Solution: A python statement that evaluates to either True or False. Typically contains a comparison operator (<, >, <=, >=, ==, !=) and may contain boolean operators such as AND, OR, or NOT.

- (c) [3 pts] conditional statement

Solution: A statement that controls the flow of execution depending on some condition. In Python the keywords if, elif, and else are used for conditional statements.

2. (10 points)

Read the following pieces of code. Determine whether they will be executed successfully or not. If yes, show the result that would be printed. If not, explain why not.

- (a) [2 pts]

```
myTuple1 = (1,2,3)
myTuple1.append(4)
print myTuple1
```

Solution: No. Because tuple is immutable, so there is not an append function for tuple.

- (b) [2 pts]

```
myTuple2 = (1,2,3)
myTuple3 = (4)
print myTuple2+myTuple3
```

Solution: No. Because myTuple3 is not a tuple but an int.

- (c) [2 pts]

```
myList=[0,3,4,1]
myList.remove(3)
print myList
```

Solution: Answer: Yes. [0, 4, 1]

(d) [2 pts]

```
myDictionary1={ a:1, b:2, c:3}
myDictionary1[d]=4
print myDictionary1
```

Solution: No. Because a, b, c, d here are neither characters nor strings. They will be treated as undefined variable names.

(e) [2 pts]

```
myDictionary2={ 10 : 1, 20 : 2, 30 : 3}
print myDictionary2[1]
```

Solution: Answer: No. Because elements in dictionary can only be accessed by keys but not indexes. In myDictionary2, 1 is not one of the keys.

3. (9 points)

Beside each expression, write “LEGAL” if the expression is legal python code, or “ERROR” if the expression would cause an error. Assume that myList, myTuple, and myString are a list, a tuple, and a string, respectively, and all contain at least one element.

1. myList = myList + [4]
2. myList.append(4)
3. del myList[0]
4. myTuple = myTuple + (4,)
5. myTuple.append(4)
6. del myTuple[0]
7. myString = myString + “4”
8. myString.append(4)
9. del myString[0]

Solution: Grading: +1 for each correct answer.

- 1) myList = myList + [4] LEGAL
- 2) myList.append(4) LEGAL
- 3) del myList[0] LEGAL

- 4) `myTuple = myTuple + (4,)` LEGAL
- 5) `myTuple.append(4)` ERROR
- 6) `del myTuple[0]` ERROR
- 7) `myString = myString + "4"` LEGAL
- 8) `myString.append(4)` ERROR
- 9) `del myString[0]` ERROR

4. (5 points)

Write a function called `removeFirst` that accepts a list as a parameter. It should remove the value at index 0 from the list. Note that it should not return anything (returns `None`). Note that this function must actually modify the list passed in, and not just create a second list with the first item removed. You may assume the list you are given will have at least one element.

For example:

```
>>> a = [1, 2, 3, 4]
>>> b = removeFirst(a)
>>> print a
[2, 3, 4]
>>> print b
None
```

Solution:

```
def removeFirst(aList):
    del aList[0]
    return None #Or a blank line.
```

Grading:

- 1 pt - Correct def statement
- 2 pt - Correctly removes first element using del keyword.
- 2 pt - returns None, either deliberately, or by default.

5. (11 points)

For each of the following questions, give a brief answer:

(a) [1 pt] What does it mean for a compound data type to be mutable?

Solution: It means that you can change items (elements) within the compound data type.

(b) [1 pt] Which compound data types are mutable?

Solution: Lists and Dictionaries are mutable.

(c) [1 pt] Which compound data types are immutable?

Solution: Strings and Tuples are immutable.

(d) [4 pts] Assume each of the following lines is entered in the IDLE shell one at a time. Circle any of the lines that produces an error. If no errors are found then write what is printed (from a and b).

```
a = [[[1, 2, 3, 4, 5], [6,7,8]], 9]
a[-1] = 200
b = a[:]
b[0][0][3] = 17
```

```
print a
print b
```

Solution:
[[[1, 2, 3, 17, 5], [6, 7, 8]], 200]
[[[1, 2, 3, 17, 5], [6, 7, 8]], 200]
Grading: +2 for each line...

(e) [4 pts] A text file named "fruits.txt" contains the following three lines:

```
Apple
Blueberry
Cantaloupe
```

The following code is run:

```
f = open("fruits.txt", "r")
f.readline()
s = f.readline()
f.close()
```

```
f = open("output.txt", "w")
f.write(s + "\n")
f.write("Orange\n")
f.close()
```

What does the file "output.txt" contain?

Solution: Answer:

Blueberry

Orange

Grading: +2 for each correct line. -1 for each incorrect line, -1 if they don't draw them vertically, or indicate the newline character.

6. (10 points)

Write a function called `studentReadLines` that accepts the name of a file as a string parameter. The function will return a list containing each individual line of the file. For example, if the file contained the following text:

```
Hello, guys!  
Good Luck on your test!  
Hope you enjoy your weekend!
```

Your function would return:

```
["Hello, guys!\n", "Good Luck on your test!\n", "Hope you enjoy your weekend!\n"]
```

Solution:

```
def studentReadLines( filename ):  
    f = open( filename, "r")  
    data = f.readlines()  
    f.close()  
    return data
```

Grading: 2 points for the header

2 points for opening the file

2 points for closing the file

2 - get the list of lines

2 - Returns the list.

-1 for hardcoding 3 lines

-1 for f.close without ()

7. (20 points)

Write a function named `runAway` that accepts an integer as a parameter. The integer corresponds to how long your function should run, in seconds, before returning.

While running, your function should get the values of the robots' IR sensors (not Obstacle sensors) and any time it detects something is behind the robot with either sensor, it should move forward at full speed until the sensor no longer detects anything. (Then the robot should stop, either until the number of seconds is up, or until something else is detected behind the robot.) When your function returns, it should return `None`.

Make sure that, no matter what, your robot stops and the function returns when the time is up! You may assume that the `myro` library is already loaded and that your robot is already connected (`init` has been called) to the computer.

Solution:

```
def runAway(time):
    while timeRemaining(time):
        if getIR("left") == 0 or getIR("right") == 0:
            forward(1)
        else:
            stop()

    stop()
```

Grading: 2 point for correct header

6 points for checking the `getIR` sensors (2 points if they only check one side of the sensors)

6 points for moving forward when either sensor is triggered (zeros)

6 points for stopping the robot and returning when the time is up.

Extra Credit Questions:

1. What is the decimal representation for the binary number $(101101)_2$ (1 pt)?
2. What is the binary representation for the decimal number $(15)_{10}$ (1 pt)?
3. What is the hexadecimal representation for the decimal number $(31)_{10}$ (1 pt)?
4. What is the name of the Braitenberg vehicle that moves away from the light (1 pt)?
5. What does CSS stand for (1 pt)?