Name : ______________________________

Grading TA: ______________________________

- 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. Do NOT sign nor take this exam if you do not agree with the honor code.
- 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 6 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.

I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Georgia Tech community. I have also read and understand the requirements outlined above.

Signature: ______________________________

Question	Points	Score
1. Vocabulary	3	
2. Multiple Choice	11	
3. In the Cups	5	
4. Valid Variables	5	
5. Slicing Strings	4	
6. Guess Number	10	
Total:	38	

1. *(3 points)*
   Using 1-2 sentences, define the vocabulary term **boolean expression**. Be brief, and to the point.

   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.

2. *(11 points)*
   For each of the following multiple choice questions, indicate the most correct answer by circling it!

   (a) [1 pt] What is printed by the following code?

```
def myRound(x):
   print ("{:.3f}".format(x))
myRound(2)
```

   A. 2.000 B. 2.00 C. 2.0 D. 2 E. 200 F. 2000

   (b) [1 pt] What value is referenced by the variable x after this code is executed?

```
x = 1
count = 0

while count < 4:
  if x % 2 == 0:
     x = x + 1
  else:
     x = x - 1
  count = count + 1
```

   A. 3 B. 2 **C. 1** D. 0 E. -1 F. -2

(c) [1 pt] Which of these commands is **NOT** a valid robot function?

A. beep(1,4400) **B. showName()** C. takePicture() D. forward(1,1)
E. turnLeft(1,2)

(d) [1 pt] What does aList reference after the following code is executed?

```
aList = [True, [1,2,3], (1,2,3)]
aList[2][0] = False
```

A. `[True, [1,2,3], (False,2,3)]`

B. `[True, [False,2,3], (1,2,3)]`

C. `[True, [1,2,3], False]`

D. This code generates an exception

(e) [1 pt] If the variable aList refers to a list, how would you make a copy of this list?

A. `bList = aList{:}`

B. `bList = aList[:]`

C. `bList = aList`

D. aList = bList[:]

(f) [1 pt] What does aList reference after the following code is executed?

```
aList = [True, [1,2,3], (1,2,3)]
aList[2] = False
```

A. `[True, [1,2,3], (False,2,3)]`

B. `[True, [False,2,3], (1,2,3)]`

C. `[True, [1,2,3], False]`

D. This code generates an exception

(g) [1 pt] Examine the following code. What would it print?

```
names = ["Harry", "Martha", "Howard", "Louis", "Nathan"]
newNamesList = names[:]
for person in names:
    if person[0] == "H":
        names.remove(person)
print(newNamesList)
```

A. `["Harry", "Howard"]`

B. `["Martha", "Louis", "Nathan"]`

C. `["Harry", "Martha", "Howard", "Louis", "Nathan"]`

D. `names`

(h) [1 pt] Examine the following code. What will b refer to once it has executed?

```
a = [’tulip’,’daisy’,’petunia’,[’rose’,’lily’]]
b = a[2:4]
```

A. `[’daisy’, ’petunia’]`

B. `[’petunia’, ’rose’]`

C. `[’petunia’, [’rose’, ’lily’] ]`

D. `[’daisy’, ’petunia’, [’rose’, ’lily’]]`

E. This code will raise an exception

(i) [1 pt] Examine the following code. What will x refer to once it has executed?

```
x = [(5,4)]
x[1] = 3
```

A. `[(5,3)]`

B. `[(5,4), 3]`

C. `[(3,4)]`

D. `[3]`

E. This code will raise an exception

(j) [1 pt] Examine the following code. How many times will the digit 1 be printed when it executes?

```
numList = ["1", 1, 2, "3"]
for x in numList:
    numString = str(x)
    numLength = len(numString)
    for y in range(0,numLength):
        print(numString)
```

A. 1 time

B. 4 times

C. 2 times

D. 3 time

(k) [1 pt] Examine the following code. What will newGreetings refer to once it has executed?

```
greetings = ["Hello", "Hi", "Hey", "Ola", "Namaste"]
greetings[1] = "Yo"
greetings[3] = "Hello"
newGreetings = greetings[0:3]
```

A. `["Hello", "Yo", "Hey", "Hello", "Namaste"]`

B. `["Hello", "Yo", "Hey", "Hello"]`

C. `["Hello", "Yo", "Hey", "Ola", "Namaste"]`

D. `["Hello", "Yo", "Hey"]`

E. `["Hello", "Hi", "Hey"]`

3. *(5 points)*
Beside the code, write down what the following code would print:

```
for x in range(1,6):
    if x < 4:
        print("cup")
    elif x > 4 and x <= 6:
        print("cake")
    else:
        print("cuppy")
```

Solution:

```
cup
cup
cup
cuppy
cake
Grading:  +1 for each correct line
(-1 for each incorrect addition or if they are not on different lines)
```

4. *(5 points)*
Examine each of the following variable assignments. If the code and variable name is valid, write ”correct” next to it. If the code or variable name is invalid, write ”invalid” next to it.

```
my_code = "dabes"

pass = "One Free"

7dwarves = 1 + 1 + 1 + 1 + 1 + 1 + 1

cashmonay$$$ = 5000

meaning42 = "completed"
```

Solution: Grading: +1 for each correct answer.

```
my_code = "dabes" CORRECT
pass = "One Free" INVALID (keyword)
7dwarves = 1 + 1 + 1 + 1 + 1 + 1 + 1 INVALID (number in varname)
cashmonay$$$ = 5000 INALID (can't have dolarsigns)
meaning42 = "completed" CORRECT
```

5. *(4 points)*
Examine the following code and write what it would print beside it:

```
aStr="Hippopotamus"
bStr=aStr[0:3]
cStr=bStr[::-1]
bStr=bStr+"po"
aStr=aStr[::-1]

print(aStr)
print(bStr)
print(cStr)
```

Solution:

```
sumatopoppiH
Hippo
piH

Grading:
 +1 for each correct line (with/without quotes) (3 total)
 +1 overall for not having quotes.
```

6. *(10 points)*
Write a function `guessNumber` that plays a number guessing game with a user. One person will call the function and give it an integer as an argument. Your function will ask another user to enter an integer. (Give them four total tries.) If the number is too high, you should print out “Too High!”, if too low, you should print out “Too Low!”, and give the user another chance to guess the number. If the user guesses the number correctly, the function should print “You got it!” and return the number of guesses the user made.

If the user does not guess the number in 4 tries, you should print “Game Over!” and also return the number of tries (4).

Example function run:

```
>>> tries = guessNumber(7)
Guess the number: 6
Too Low!
Guess the number: 10
Too High!
Guess the number: 7
You got it!
>>> tries
3
```

Solution:

```
def guessNumber( number ):
    userNum = int( input("Guess the number:") )
    tries = 1
    while userNum != number:
        if userNum > number:
            print("Too high!")

        elif userNum < number:
            print("Too low!")

        if tries > 4:
            print("Game Over")
            return tries

        userNum = int( input("Guess the number") )
        tries = tries + 1
    print("You Got it!")
    return tries
```

Grading:
+1 for a correct header.
+2 for asking user input and making it an int
+2 for correctly asking up to 4 times (looping)
+2 for ending the function after the 4th try
+2 for correct conditionals on number too high or low +1 for printing winning statement and returning number tries on correct num