Name: _____ Andrew Id: _____

15-112 Spring 2019 Quiz 9

Up to 25 minutes. No calculators, no notes, no books, no computers. Show your work!

1. (20 points) For each question, fill in the circle for of the most correct answer.

   (a) What is a Class in Python?
   - ◯ A template
   - ◯ A specific item
   - ◯ A specialized function
   - ◯ An inheritance

   (b) What is an Instance in Python?
   - ◯ A template
   - ◯ A specific item
   - ◯ A specialized function
   - ◯ An inheritance

   (c) What does it mean for a class to override a method?
   - ◯ It gets the method from its superclass
   - ◯ It gets the method from its subclass
   - ◯ It calls a method from the superclass using `super()`
   - ◯ It changes how the method works from the original version

   (d) Which of the following can be a superclass of Student?
   - ◯ University
   - ◯ ComputerScienceStudent
   - ◯ Course
   - ◯ Person

   (e) What is the difference between `__repr__(self)` and `__str__(self)`?
   - ◯ `__str__` is used when the object is converted to a string, `__repr__` is used when the object is written to a file
   - ◯ `__str__` is used when the object is directly converted to a string, `__repr__` is used when the object is indirectly converted to a string
   - ◯ `__str__` is used when the object is converted to a string, `__repr__` is used when the object is hashed
   - ◯ There is no difference.

2. (25 points) **Free Response:** Write the non-destructive function `removeVowels(s)` which takes a string, `s`, and returns a copy of it with all of the vowels removed. For example, `removeVowels("Hello there")` returns `"Hll thr"`.

   This function must be written recursively. A solution that uses loops, comprehensions, generators, or iterative built-in functions such as range will receive no credit.

3. (20 points) **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box to the right of the code.

```
def ct(lst, depth=0):
    print(depth, "in:", lst)
    if len(lst) == 0:
        result = []
    elif lst[0]%2 == 0:
        result = ct(lst[1:], depth+1)
    else:
        result = [lst[0]]
        result += ct(lst[1:], depth+1)
    print(depth, "out:", result)
    return result

ct([7, 40, 33])
```

4. (35 points) **Free Response:** Write the classes Building and School so that they pass the following test cases. You may not hardcode any test cases. For full credit you must use inheritance appropriately.

```
# A building is either open or closed.  It starts out open.
b = Building()
assert(str(b) == "Building(Closed=False)")
# A building can be closed and opened
b.close()
assert(str(b) == "Building(Closed=True)")
b.open()
assert(str(b) == "Building(Closed=False)")

# A School is a Building that has students.
s = School(2)
assert(str(s) == "School(Students=2,Closed=False)")
# You can't close a school if there are students inside
ok = False
try:
    s.close()
except:
    ok = True
assert(ok)
# You can remove students, but the number of students inside can't go
# below 0.
s.removeStudent()
assert(str(s) == "School(Students=1,Closed=False)")
s.removeStudent()
assert(str(s) == "School(Students=0,Closed=False)")
s.removeStudent()
assert(str(s) == "School(Students=0,Closed=False)")
# Once there are no students, then you can close the school.
s.close()
assert(str(s) == "School(Students=0,Closed=True)")

# Check various things about inheritance
assert(isinstance(b, Building) == True)
assert(isinstance(b, School) == False)
assert(isinstance(s, Building) == True)
assert(isinstance(s, School) == True)
# Buildings don't have students....
ok = False
try:
    b.removeStudent()
except:
    ok = True
assert(ok)
```

Write your solution on the next page.

Your solution to question 4 goes here.