

fullName:_____andrewID:_____

recitationLetter:_____

15-112 N22

Quiz2 version A [40 min.]

You **MUST** stop writing and hand in this **entire** quiz when instructed in lecture.

- You may not unstaple any pages.
- You may not use your own scrap paper. If you must use additional scrap paper, raise your hand and we will provide some. You must hand this in with your paper quiz, and we will not grade it.
- Failure to hand in an intact quiz will be considered cheating. Discussing the quiz with anyone in any way, even briefly, is cheating. (You may discuss it only once the quiz has been posted to the course website.)
- You may not use any concepts (including builtin functions) we have not covered in the notes this semester.
- You may not use lists, list or tuple indexing, dictionaries, sets, or recursion.
- We may test your code using additional test cases. Do not hardcode.
- We do not deduct points for bad style on quizzes
- Assume `almostEqual(x, y)` and `roundHalfUp(n)` are both supplied for you. You must write all other helper functions you wish to use.

True or False [2pts ea]

Write only the whole word "True" or "False" (and not just T or F).

_____ **TF1:** The Model contains all the data we need for the animation, which we store in the app object's attributes.

_____ **TF2:** Controllers can only update the Model, they cannot update the View.

_____ **TF3:** The View should never update the Model.

Short Answer [2pts ea]

Write the word Model, View, or Controller to indicate what part of MVC best describes each of these features.

_____ **SA1:** The values `app.width` and `app.height`

_____ **SA2:** A circle drawn on the canvas

_____ **SA3:** The `mousePressed` function

_____ **SA4:** The `timerFired` function

CT1: Code Tracing [10pts]

Indicate what the following code prints. Place your answers (and nothing else) in the box below.

```
def ct1(s):
    r = ""
    for i in range(len(s)):
        c = s[i]
        if c.isupper():
            c = chr(ord(c) + i)
        elif c.islower():
            c = chr(ord(c) + 2)
            c = c.upper()
        r = c + r
    return r

print(ct1("catDAD?"))
```

CT2: Code Tracing [12pts]

Indicate what the following code prints. Place your answers (and nothing else) in the box below.

```
def ct2(s):  
    r = ""  
    while (len(s) > 1):  
        r += s[:1] + s[2:4] + "-"  
        s = s[1:len(s)-1:2]  
        print(s)  
    return r + s  
  
print(ct2("abcd123"))
```

Free Response 1: collapseWhitespace(s) [30pts]

Without using the `s.replace()` method, write the function `collapseWhitespace(s)` that takes a string `s` and returns an equivalent string except that each occurrence of whitespace in the string is replaced by a single space. So, for example, `collapseWhitespace("a\t\t\tb\n\n c")` replaces the three tabs with a single space, and the two newlines with another single space, returning `"a b c"`.

Here are some test cases:

```
assert(collapseWhitespace("a\nb") == "a b")
assert(collapseWhitespace("a\n \t b") == "a b")
assert(collapseWhitespace("a\n \t b \n\n \t\t\t c ") == "a b c ")
assert(collapseWhitespace("\ta\nb") == " a b")
assert(collapseWhitespace("15 112 \n") == "15 112 ")
assert(collapseWhitespace("\t \n") == " ")
assert(collapseWhitespace("") == "") # empty string
assert(collapseWhitespace(" ") == " ") # single space string
```

You may continue your FR1 answer here, if you wish

Free Response 2: Dot Animation [34pts]

Write an animation with the following features:

1. A large blue circle with radius 70 is located in the center of the canvas
2. A small falling circle with radius 20 is initially located with its center at the top of the canvas, horizontally centered, and colored red
3. Every 100ms the small falling circle moves down 20 pixels
4. The color of the small falling circle becomes green whenever it is touching the large blue circle. Whenever the circles are not touching, the small falling circle is red.
5. Note that the small circle should remain visible when it passes in front of the large circle.
6. Whenever the center of the small circle reaches the bottom of the canvas, it wraps around to the top of the canvas
7. Whenever the large blue circle is clicked with the mouse, the small falling circle immediately moves to the top of the canvas. Other mouse presses should be ignored.

Note: If you wish, you may abbreviate app, event, and canvas as a, e, and c

```
from cmu_112_graphics import *  
#Assume we will call runApp() after your code  
# with reasonable width and height values
```

You may continue your FR2 answer here, if you wish

You may continue your FR2 answer here, if you wish

bonusCT: Code Tracing [2pts]

This question is optional. Indicate what the following code prints. Place your answers (and nothing else) in the box below.

```
def bonusCt(s):
    t = ''
    for i in range(len(s)):
        t += s[i:] + s[i:][:-1]
    for c in s[::-1]:
        t = (t.replace(c+c, '')) +
            str(t.count(c) - 2)
    return t
print(bonusCt('abcd'))
```