

1. _____/18

2. _____/15

3. _____/15

4. _____/12

5. _____/25

6. _____/12

7. _____/1

8. _____/2

Athena User Name-----
Recitation hour

Total _____/100

This quiz is open book and open notes, but do not use a computer.

Please **write your name on the top of each page**, and your user name and the hour of the recitation you attend on the first page. Answer all questions in the boxes provided.

1) Are each of the following True or False (18 points)

☐ 1.1. “Glass box testing” and “black box testing” mean the same thing.☐ 1.2. If a program deals in only immutable types, aliasing is irrelevant.☐ 1.3. When applied to the same data, a bisection search will never take longer to run than a linear search.☐ 1.4. There are algorithms that can be implemented with `for` loops but not with `while` loops.☐ 1.5. When Newton-Raphson is used to find a root of a polynomial, the **order** of complexity is quadratic in the degree of the polynomial.☐ 1.6. The statement `d = {1:1, []:2}` is legal in Python.

2) What does the following code print? Note that there is a print statement in implementation of `f`. (15 points)

```
def f(s, i, j):  
    if s[i] != s[j]:  
        s = s[j] + s[i+1:j-1] + s[i]  
    if j - i > 2:  
        print s  
        return f(s, i+1, j-1)  
    else:  
        return s  
  
L = 'abbca'  
print f(L, 0, len(L) - 1)
```

3) Consider the function definition of `squareRootBi` shown here. This code fails to meet its specification. 1) Give a value of `x` for which it will fail. 2) show how to correct the code so that it will not fail. (15 points)

```
def squareRootBi(x):  
    """Assumes x is a float > 0  
       Return y s.t. y <= 0 and y*y is within 0.01 of x"""  
    epsilon = 0.01  
    low = -x  
    high = 0  
    guess = low  
    while abs(guess**2 - x) > epsilon:  
        if guess**2 > x:  
            low = guess  
        else:  
            high = guess  
        guess = (low + high)/2.0  
    return guess
```

4) What does the following print? (12 points)

```
a = (1, 2, [])  
b = a  
c = {1:1, 2:2}  
d = c  
a = a + ('a',)  
c[2] = 3  
print a[0] == c[1]  
print c == d  
print b[-1]  
print c.keys().sort()
```

5) Write a Python function that meets the following specification. (25 points)

```
def g(L):  
    """assumes L is a non-empty list of ints  
    return an x s.t. x is in L and no other int  
    occurs more often than x in L."""
```

6) Consider the following code:

```
def f(x, s):  
    if x == 0 or len(s) == 0:  
        return max(x, len(s))  
    for c in s:  
        return f(x - 1, '')  
  
x = 101  
print f(x, str(x))
```

6.1. What does it print? (6 points)

6.2. Which of the following provides the tightest bound on the complexity of f? (6 points)

- A. $O(x)$
- B. $O(\log_{10}(x))$
- C. $O(\text{len}(s))$
- D. $O(1)$

7) Do you think that the lectures are too slow paced, too fast paced, about right? (1 point)

Too slow 1 2 3 4 5 Too fast

8) Do you think that the problem sets are too short, too long, about right? (2 points)

Too short 1 2 3 4 5 Too long