

Full Name: **Answer Key**

Email Address (UW Net ID): _____@uw.edu

Section: _____

CSE 160 Winter 2024 - Midterm Exam

Instructions:

- You have **45 minutes** to complete this exam.
- The exam is **closed book**, including no calculators, computers, phones, watches or other electronics.
- You are allowed a single sheet of notes for yourself.
- We also provide a syntax reference sheet.
- Turn in *all sheets* of this exam, together and in the same order when you are finished.
- When time has been called, you must put down your pencil and stop writing.
 - **Points will be deducted if you are still writing after time has been called.**
- You may only use parts and features of Python that have been covered in class up to this point.
- You may ask questions by raising your hand, and a TA will come over to you.

Good luck!

Question	Points
Question 1	6
Question 2	4
Question 3	6
Question 4	5
Question 5	5
TOTAL	26

Question 1) (6 points) For each expression on the left, write the value that x evaluates to and then the type of that value. If the expression results in an Error, write "Error" in both columns. (You do not have to explain what the error is.)

For example:

<i>[Example]</i> Expression	<i>[Example]</i> Value of x	<i>[Example]</i> Type of x
x = 3	3	int
x = 3..0	Error	Error

Expression	Value of x	Type of x
x = 10 * 3.7	37.0	float
lst = [4, 3, 2, 1] x = lst.reverse()	None	NoneType
x = (67.3 * 10) + 3 > 65	True	bool
x = 2 ** 3 // 4	2	int
x = 'Please don't make this mistake'	Error	Error
a = [6, 5, 4, 3, 2, 1, 0] b = "success" c = b[a[-3]] + b[::4] d = a[1] * a[2] * a[-3] * a[4 // 2] x = str(c) + str(d)	"cse160"	string

Question 2) (4 points) For each of the code snippets on the left, write the corresponding output on the right.

Code	Output
<pre>total_sum = 0 for i in range(2, 8, 2): total_sum = total_sum + i print(total_sum)</pre>	<p>12</p>
<pre>for i in range(2): for j in range(2): print(i + j)</pre>	<p>0 1 1 2</p>
<pre>month = "January" day = 10 output_string = "" if day <= 10: output_string += "Early " if day >= 20: output_string += "Late " else: output_string += "Mid " output_string += month print(output_string)</pre>	<p>Early Mid January</p>
<pre>hours = 0 day = True if hours < 12 and day: hours += 12 elif hours == 12: day = False else: hours -= 6 print(day) print(hours)</pre>	<p>True 12</p>

Question 3) Consider the following two functions:

```
def my_function(a, b):
    result = a - b
    return result
```

```
def loop(x, y):
    for i in range(3):
        if i % 2 == 0:
            x = my_function(x, y)
        else:
            y = my_function(y, x)
    return x + y
```

Fill in the tables below such that the code in the "Call to loop()" column produces the output in the "Output" column. The code in the "Call to loop()" column *must* use the loop function defined above.

For example:

[Example] Code	[Example] Output
<code>print("2 cubed is", 2 ** 3)</code>	2 cubed is 8

a) (3 points) Write the output produced by the code on the left:

Call to loop()	Output
<code>print(loop(4, -2))</code>	6
<code>print(loop(-1, -2))</code>	1
<code>x = 3 y = 2 result = loop(y, x) print(result)</code>	-1

b) (3 points) Write a call to the function loop() that will produce the value on the right. (Hint: you may find it helpful to "unroll" the loop.)

Call to loop()	Output
loop(8, 0)	8
loop(0, 8)	-8
loop(1, 1)	0

Question 4) You are given a file named `start.txt` which contains the following text:

`start.txt`

```
Knock knock
Owls say who?
Yes they do
```

- a) (2 points) Write the contents of `output_a.txt` after running the following program (you may assume it is in the same directory as `start.txt`).

```
start_file = open('start.txt', 'r')
output_a = open('output_a.txt', 'w')
for line in start_file:
    for word in line.split():
        output_a.write(word + '\n')
start_file.close()
output_a.close()
```

`output_a.txt`

```
Knock
knock
Owls
say
who?
Yes
they
do
```

- b) (3 points) Write the contents of `output_b.txt` after running the following program (you may assume it is in the same directory as `start.txt`).

```
start_file = open('start.txt', 'r')
output_b = open('output_b.txt', 'w')
for line in start_file:
    output_b.write(line + '\n')
for line in start_file:
    output_b.write(line + '\n')
start_file.close()
output_b.close()
```

`output_b.txt`

```
Knock knock
Owls say who?
Yes they do
```

Question 5 (5 points) Instead of analyzing DNA (A, T, C, G) sequences like in HW2, now you are given an RNA (A, U, C, G) sequence as an input string.

Each three nucleotides in the sequence translates to an amino acid. Amino acids are the building blocks to make proteins! Use a **for loop**, **range**, and **string slicing** to "translate" the RNA sequence to the amino acids. Every three nucleotides corresponds to a single amino acid.

Specifically, your objective is to look for the following amino acids:

RNA to target Amino Acid Mapping:

"AUG" --> "Methionine"

"UGC" --> "Cysteine"

"UCU" --> "Serine"

For example:

given

```
rna = "AUGCUC AUG"
```

your code should print out:

```
['Methionine', 'Methionine']
```

given:

```
rna = "ACCUUUAUGAUUUUGCUACCAAUUUUUUGCCGAUCUGCAUCUUUUUGGG"
```

your code should print out:

```
['Methionine', 'Cysteine', 'Serine', 'Cysteine', 'Serine', 'Serine']
```

```
def working2(rna):  
    n = len(rna)-1  
    # Initialize list for amino acid chain  
    aa_chain = []  
  
    # For loop iterates over RNA sequence using a step size of 3  
    for i in range(0, n, 3):  
        # set codon variable to slice three nucleotides per codon  
        codon = rna[i:i+3]  
        # Check each codon to see if the correct amino acid is present  
        # if so, append it to the amino acid list  
        if codon == "AUG":  
            aa_chain.append("Methionine")  
        elif codon == "UGC":  
            aa_chain.append("Cysteine")  
        elif codon == "UCU":  
            aa_chain.append("Serine")  
  
    return aa_chain
```

Extra Credit (1 point): Choose one:

- 1) Write the names of one or two TAs.
- 2) Draw what you think the CSE 160 mascot should be.