Instructions:

1. You are allowed to bring one index card of notes (3in x 5in). Do not submit the index card with the exam.

2. This is a closed book exam. Do not confer with any other person.

3. Computers are not allowed.

4. Calculators are not allowed.

5. Show your work. Partial credit will be given. Grading will be based on correctness, clarity and neatness.

6. I suggest that you read the whole exam before beginning to work any problem. Budget your time wisely.

7. All the programs in the problems must be written in Python.

8. The duration of this exam is 75 minutes.

**DO NOT BEGIN THE EXAM UNTIL INSTRUCTED TO DO SO. GOOD LUCK!**

Please sign the academic integrity statement:

"On my honor, as an Aggie, I have neither given nor received unauthorized aid on this academic work. In particular, I certify that I have not received or given any assistance that is contrary to the letter or the spirit of the guidelines for this exam."

Student Signature: ____________________________________________
1. (6 points)
   (a) List one difference between a compiler and an interpreter.
       Interpreter:
       Translates program one statement at a time
       No intermediate object code is generated
       Continues translating the program until the first error occurs
       Compiler:
       Scans the entire program and translates it as into machine code
       Generates intermediate object code
       It generates the error message only after scanning the whole program
   (b) What are the numeric data types supported in Python?
       Python supports three numeric data types: int (signed integers)
       float (floating point real numbers)
       complex (complex numbers): a + bj
   (c) What is an immutable object in Python?
       An immutable object in Python is an object that cannot be modified or changed.

2. (8 points) Complete the following statements by circling the right answer:
   (a) The float() function returns a float object from some numbers or strings. This operation is: ___
       i. a logical operation
       ii. an explicit type conversion
       iii. a concatenation
       iv. an implicit type conversion
       v. an enumeration
   (b) What is the data type of the following Python expression: 1 + int(3.5)/2:
       i. a string
       ii. an integer
       iii. a boolean
       iv. a float
       v. a division
   (c) Which of the following statements produces an error?
       Assume string_1 = "abc" and string_2 = "123":
       i. string_2 = string_1
       ii. string_1 = string_2 + str(456)
       iii. print(string_1 + string_2)
       iv. string_1[2] = "B"
   (d) We cannot delete or remove characters from a string because: ___
       i. strings are iterables
       ii. strings are objects
       iii. strings are mutable
       iv. strings are not objects
       v. strings can include white-space characters
3. (10 points) Find the lines of code that are incorrect in this program.

- Write down the incorrect line numbers.
- Write a brief description of the error next to each incorrect line number.

```python
university = "TAMU"
title = "Programming"
prefix = "CSCE"
number = 110
credits = 3

lecture = prefix + str(number)
course = title + prefix + number
letters = len(university)
number = prefix * letters
title = title[::-1]
title[1:4] = "Introduction"
extra = input("Number of extra credits: ")
total = credits + extra
```

- Line 8: We can only concatenate strings, not integers and strings.
- line 12: A string is immutable
- line 14: Unsupported operand + for integer and string
4. (14 points) Consider the following program.

```
1 pi = 3.14
2 a = 23
3 b = "code"
4 c = int(pi)
5 d = 2
6 print (a % 5)    # Line A
7 print (a // 5)   # Line B
8 print ((c >= pi) and (c > d))   # Line C
9 print ((a == d) or (a != 110))  # Line D
10 print ("d" in b)  # Line E
11 print (not ("ce" in b))        # Line F
12 print (f"number: {d**3}" )   # Line G
```

(a) (2 points) What is the output of Line A? 3  
(b) (2 points) What is the output of Line B? 4  
(c) (2 points) What is the output of Line C? False  
(d) (2 points) What is the output of Line D? True  
(e) (2 points) What is the output of Line E? True  
(f) (2 points) What is the output of Line F? True  
(g) (2 points) What is the output of Line G? number: 8

5. (14 points) Consider the following program.

```
x = "code"
y = "software"
print(x[3])   # Line A
print(x[2:] + x[:2])   # Line B
print(x[3] + y[-3])   # Line C
print(y[4:])   # Line D
print(x[1:4])   # Line E
print(y[8 - len(x)])   # Line F
print(y[1::3])   # Line G
```

(a) (2 points) What is the output of Line A? e  
(b) (2 points) What is the output of Line B? deco  
(c) (2 points) What is the output of Line C? ea  
(d) (2 points) What is the output of Line D? ware  
(e) (2 points) What is the output of Line E? ode  
(f) (2 points) What is the output of Line F? w  
(g) (2 points) What is the output of Line G? owe
6. (8 points)

Write a python program that asks a user for two numbers \(a\) and \(b\) in any order.
The program should print all the numbers between \(a\) and \(b\) that are even and multiple of 3.
- Include \(a\) if \(a\) is even and multiple of 3.
- Include \(b\) if \(b\) is even and multiple of 3.
- All the numbers must be printed on the same line.

Sample output:

```
low = int(input("a: "))
high = int(input("b: "))

if low > high:
    low, high = high, low

print(f"numbers: ", end="")
for n in range(low, high + 1):
    if n % 2 == 0 and n % 3 == 0:
        print(f"{n}", end=" ")
```

Sample output:

```
Numbers: 6 12 18 24 30 36 42 48 54 60
```

Sample output:

```
Numbers: 12 18 24
```

Sample output:

```
Numbers: 12 18 24 30
```
7. (6 points)

Write a Python program that asks a user for a number n. The program should count and print the number of even and odd numbers between 0 and n (n included).

Sample output:

```
n: 10
There are 6 even numbers between 0 and 10
There are 5 odd numbers between 0 and 10
```

Sample output:

```
n: 15
There are 8 even numbers between 0 and 15
There are 8 odd numbers between 0 and 15
```

```
n = int(input("n: "))
count_odd = 0
count_even = 0
for x in range(n + 1):
    if x % 2 == 0:
        count_even += 1
    else:
        count_odd += 1
print(f"There are {count_even} even numbers between 0 and {n}\")
print(f"There are {count_odd} odd numbers between 0 and {n}\")
```
8. (10 points) Write a Python program which asks a user for a word and performs letters manipulation.

- If the word is empty, the program should print the message *empty!*
- Otherwise, if the word has between one and four letters, the program should print the reversed word.
- Otherwise, the program should print a word made of the first two and the last two letters of the original word.

Sample output:

```python
Enter a word: 
Result: empty!
```

Sample output:

```python
Enter a word: bio
Result: oib
```

Sample output:

```python
Enter a word: memory
Result: mery
```

```python
word = input("Enter a word: ")
if len(word) == 0:
    print(f"Result: empty!")
elif len(word) > 0 and len(word) <= 4:
    print(f"Result: {word[::-1]}")
else:
    print(f"Result: {word[0:2] + word[-2:]}")
```
9. (13 points) Consider the following program.

```python
x = 11  # Line A
y = 7

if x < 3:
    y = y + 1
else:
    x = x - 2
    if x <= 3:
        y += 3

if x == y:
    x = x + 1
if x != y:
    x = x + 1

print(x)  # Line B
print(y)  # Line C
```

(a) (3 points) Write the execution order of the lines in this program. 1, 2, 4, 6, 7, 8, 11, 14, 15, 17, 18

(b) (2 points) What is output of Line B? 10

(c) (2 points) What is output of Line C? 7

(d) (3 points) Replace Line A with `x = 9`. What is the output of the program? 9 7

(e) (3 points) Replace Line A with `x = 5`. What is the output of the program? 4 10
10. (6 points)
What is the expected output for this program?

```python
n = 4
f = 1
if n < 0:
    print("n must be positive")
elif n == 0:
    print(1)
else:
    for i in range(1, n + 1):
        f = f * i
    print(f)
```

11. (5 points)
What is the expected output for this program?

```python
for i in range(6, 8):
    for j in range(3):
        if (j == 1):
            continue
        print(f"{i},{j}")
```

Output:
- 1
- 2
- 6
- 24
Extra credit (3 points)

Write a Python program which asks a user for a string. The program should add "ing" at the end of the string.

- The length of the string should be at least 3.
- If the given string already ends with "ing" then add "ly" instead.
- If the string length of the given string is less than 3, leave it unchanged.

Sample output:

1. Enter a word: car
   Result: caring

Sample output:

1. Enter a word: spring
   Result: springly

```python
word = input("Enter a word: ")
if len(word) > 2:
    if word[-3:] == "ing":
        word += "ly"
    else:
        word += "ing"
print(f"Result: {word}"")
```