CSE-101, Introduction to Programming
Midterm Exam, 2018

Name: _____ Marks: 25
Roll Number:_____ Time: 60 minutes
Section: _____
Group: _____

Instructions:
1. You will be expected to write Python code in this exam. We recommend that you draw vertical lines to make your indentation clear.
2. Assume the use of Python3 in all of the questions below.
3. Write your details on both the question paper and the answer sheet. Only the answers written in the answer sheet will be evaluated.
4. There are 2 sheets in this question paper printed both sides. There are a total of 4 questions and Q1 and Q2 have subparts. Mention the question number and the subpart number clearly.

Question 1: Write the output of the following programs:

a. (4 Marks)

```
x = 'Intro to programming' < = 'intro to programming'
y = not True and False or not False
z = 2**-3**3 >= -512
print('x= ' + str(x))
print('y= ' + str(y))
print('z= ' + str(z))
if (x and y and z):
    x= y > z
    print ('x='+ str(x))
else:
    x = y and z
    print ('x='+ str(x))
```

Ans:
x= True
y= True
z= True
x=False

b. (1 Mark)

```
x = "foo"
y = "bar"
print (x[-1:1]*2 + y[1:-1]*4) # Use '_' for one space
```

Ans:
aaaa

Question 2: Please read the question carefully and answer the following:

a) (4 Marks)

Consider the code given below. What should the values for variables ***start***, ***end***, ***first*** and ***last*** be assigned at the beginning of this code if the required output is:

15 20 25 30
18 24 30 36
21 28 35 42

```
#variable assignment lines
start = _______
end = ________
first = ________
last = _______
m = start+5
while m  <= end-1:
        n= first
        while n < last:
            print ( " %5d " %(m*n), end = ' ')
            n += 1
        m += 1
        print()
```

Ans:
start = 0
end = 8
first = 3
last = 7

b) (2 Marks)

Can we assign ***integer*** values to variables x and y so that the string "CSE101" is printed out. If yes, what are those values for x and y?

```
x=_______
y=_______
if (x>=1) and (0<=y<=3):
    print("CSE103")
elif not(y<=4 or y>=1):
    print("CSE102")
elif x<=1 or x>=3:
    print("PSY03")
else:
    print("CSE101")
```

Ans:

x=2

y can take all integral values except 0,1,2,3

c) (2 Marks)

What would be the output of the function call: `method_3c(1325476)?`

```
def method_3c(number):
    x= number
    count = 0
    while (x > 0):
        x= int(x/10)
        count += 1
    i = 2
    while ( i < int(count/3)):
        number = int(number /10)
        i += 2
        count= count+1
    return number%10
```

Ans:

6

d) (2 Marks)

Consider the following script:

```
k=10
while k <=100:
    print(k)
    k= k + 5
```

Write an equivalent script that makes effective use of a for-loop instead of while loop.

Ans:

```
for k in range(10, 101, 5):
    print(k)
```

Question 3: (5 Marks) Write a function isNumberOkay(number) that returns True if the parameter number $a_1a_2a_3a_4....a_n$ of ***n*** digits satisfies the following pattern and False otherwise:

- Number $a_1a_2a_3a_4....a_n$ is a positive integer, and
- For all 1 <= i <= int(n/2), both a_i and a_{n-i+1} are either both even or odd.

```
def isNumberOkay(number):
""" Returns: True if number satisfies following two conditions
1. number a1a2a3a4....an is a positive integer, and
2. For all 1 <= i <= int(n/2), both ai and an-i+1 are either both
even or odd.
Example: Inputs 41770, 30387, 7777, 6752 will return True.
Precondition: number is an integer object with at least 2
digits. """
```

Ans:

```
def isNumberOkay(number):
        num=str(number)
        numdig= len(num)
        if number < 0 :
                return False
        i=1
        while i<=int(numdig/2) :
                if int(num[i-1])%2 != int(num[numdig-i])%2 :
                        return False
                i=i+1
        return True
```

Question 4: (5 Marks) In the US, 10-digit telephone numbers are typically represented in one of the two following styles:
“Parenthetical”: (555) 666-1110
“Dashed”: 555-666-1110
There is no whitespace in a Dashed phone number: they are all exactly 12 characters long. There is only one space in a Parenthetical phone number, and it is after the “)”; they are all exactly 14 characters long, counting the space.
Implement the following function according to its specification.

```
def phone_to_paren(s):
""" Returns: a string representing the phone number s in
Parenthetical form.
Precondition: s is a non-empty string that *would* be a valid
Dashed phone number EXCEPT that it possibly has spaces around
the dashes.
Examples of valid input:
555-666-1110
555 - 666 - 1110
555 - 666-1110
,etc., ... all yield the same output:
(555) 666-1110
"""
```

Ans:

```
def phone_to_paren(s):
    phno = str(s)
    if len(phno) >= 12:
        i= 0
        out= '('

        for c in phno:

            if c.isdigit() == True :
                out = out + c
                i= i+1

            if i == 3:
                out = out + ') '
                i= i+1
```

```
            if i == 7:
                out = out + '-'
                i= i+1

            if i == 14:
                break

        return out

    else:

        out = "Invalid Input"
        return out
```

CSE-101, Introduction to Programming
Midterm Exam, 2018

Name: GAURAV KHURANA
Roll Number: 2018142
Section: A
Group: 6

Marks: 25
Time: 60 minutes

Instructions:

1. You will be expected to write Python code in this exam. We recommend that you draw vertical lines to make your indentation clear.
2. Assume the use of Python3 in all of the questions below.
3. Write your details on both the question paper and the answer sheet. Only the answers written in the answer sheet will be evaluated.
4. There are 2 sheets in this question paper printed both sides. There are a total of 4 questions and Q1 and Q2 have subparts. Mention the question number and the subpart number clearly.

Question 1: Write the output of the following programs:

a. (4 Marks)

```
x = 'Intro to programming' < = 'intro to programming'
y = not True and (False or not False)
z = 2**-(3**3) >= -512
print('x= ' + str(x))
print('y= ' + str(y))
print('z= ' + str(z))
if (x and y and z):
    x= y > z
    print ('x='+ str(x))
else:
    x = y and z
    print ('x='+ str(x))
```

b. (1 Mark)

```
x = "foo"
y = "bar"
print (x[-1:1]*2 + y[1:-1]*4) # Use '_' for one space
```

Question 2: Please read the question carefully and answer the following:

a) (4 Marks)

Consider the code given below. What should the values for variables ***start***, ***end***, ***first*** and ***last*** be assigned at the beginning of this code if the required output is:

15 20 25 30
18 24 30 36
21 28 35 42

```
#variable assignment lines
start = 0
end = 8
first = 3
last = 7
m = start+5
while m  <= end-1:
        n= first
        while n < last:
            print ( " %5d " %(m*n), end = ' ')
            n += 1
        m += 1
        print()
```

b) (2 Marks)

Can we assign ***integer*** values to variables x and y so that the string "CSE101" is printed out. If yes, what are those values for x and y?

```
x=_____
y=_____
if (x>=1) and (0<=y<=3):
    print("CSE103")
elif not(y<=4 or y>=1):
    print("CSE102")
elif x<=1 or x>=3:
    print("PSY03")
else:
    print("CSE101")
```

y>4 and y<1

x=2 y=0

Question 3: (5 Marks) Write a function isNumberOkay(number) that returns True if the parameter number $a_1a_2a_3a_4....a_n$ of n digits satisfies the following pattern and False otherwise:

- Number $a_1a_2a_3a_4....a_n$ is a positive integer, and
- For all 1 <= i <= int(n/2), both a_i and a_{n-i+1} are either both even or odd.

```
def isNumberOkay(number):
""" Returns: True if number satisfies following two conditions
1. number a1a2a3a4....an is a positive integer, and
2. For all 1 <= i <= int(n/2), both ai and an-i+1 are either both
even or odd.
Example: Inputs 41770, 30387, 7777, 6752 will return True.
Precondition: number is an integer object with at least 2
digits. """
```

Question 4: (5 Marks) In the US, 10-digit telephone numbers are typically represented in one of the two following styles:
"Parenthetical": (555) 666-1110
"Dashed": 555-666-1110
There is no whitespace in a Dashed phone number: they are all exactly 12 characters long. There is only one space in a Parenthetical phone number, and it is after the ")"; they are all exactly 14 characters long, counting the space.
Implement the following function according to its specification.

```
def phone_to_paren(s):
""" Returns: a string representing the phone number s in
Parenthetical form.
Precondition: s is a non-empty string that *would* be a valid
Dashed phone number EXCEPT that it possibly has spaces around
the dashes.
Examples of valid input:
555-666-1110
555 - 666 - 1110
555 - 666-1110
,etc., ... all yield the same output:
(555) 666-1110
"""
```

c) **(2 Marks)**

What would be the output of the function call: method_3c(1325476)?

```
def method_3c(number):
    x= number
    count = 0
    while (x > 0):
        x= int(x/10)
        count += 1
    i = 2
    while ( i < int(count/3)):
        number = int(number /10)
        i += 2
        count= count+1
    return number%10
```

d) **(2 Marks)**

Consider the following script:

```
k=10
while k <=100:
    print(k)
    k= k + 5
```

Write an equivalent script that makes effective use of a for-loop instead of while loop.

CSE-101, Introduction to Programming
Midterm Exam, 2019

Marks: 20 **Time: 60 minutes**

Name: _____ **Roll Number: _____**

Section: _____ **Group: _____**

Instructions:

1. Assume the use of Python3 in all of the questions below.
2. All questions are mandatory.
3. No doubts will be discussed during the exam.

Q1) [1 + 1 = 2 Points] Code:

```
x = "IP A"                                      #1
x[-1] = "B"                                  #2
y = "EASY_CSE101-SecB"                          #3
y[-2] = "C"                                     #4
y[-3] = "E"                                     #5
print(y[5:-8]+x[-2:1]+x[0:2]+y[12:-1]+x[2:])#6
```

i) [1 Point] Indicate the lines which will throw error. Explain why.

ii) [1 Point] Write the output of Line 6 if all the error prone lines are removed.

Q2) [1 + 1 + 1 = 3 Points] Write the output of the following? In case of error, what error the program does have?

a. Code:

```
x = "CSEIP" + 1
print(x)
```

b. Code:

```
x = "CSEIP"
print(x[5])
```

c. Code

```
x = "123456"
print()
y = "abcdef"
print(y[int(x[2])])
```

Q3) [1 + 2 = 3 Points] Code:

```
a = 2
b = 1
def f1(a, b):
      c = a * b
      c = c - 5
      c = c / 3
      print(type(c))
      return c

def f2(a, b):
      b = b * 3
      a = a + b * 4 + 4
      b = 100
      d = a / 5
      e = b // 5
      return e

if a == 2:
      print(f1(a, b))
if b == 1:
      print(f2(a, b))
```

i) [1 Point] How many output lines will show up when the code is run on python console?

ii) [2 Point] What is the output?

Q4) [2 Points] What will be the output if the below code is run?

Code:

```
a = 1
b = 2
c = 3
d, e = 4, 5
if a == 1:
      print("Yes")
if b == 2:
      print("No")
elif c == 3:
      print("YesNo")
if d == 4:
      print("Yes")
if e == 5:
      print("YesYes")
else:
      print("NoNoNo")
```

Q5) [2 Points] Write the output of the following code.

```
class Point:
   def __init__(self,a,b):
      self.x=a
      self.y=b


a = Point(1,2)
def f(a):
      a.x=a.x + 1
      print("f() ", a.x)
      return a
def g(a):
    f(a)
    a = 2
    a += -1
    print('g() ', a)
    return a
def h(a):
      a.y = a.y + 3.0
      print('h() ', a.y)
      a = f(a)
      a.y += g(a)
      return a

final = h(a)
print(final.x)
print(final.y)
```

Q6) [2 + 1 = 3 Points] Fill in the blanks:

i) [1 + 1 = 2 Points] Complete the function redact so that it meets the specifications.

```
def redact(s):

 """Returns: a copy of string s where
all but the first and last letter have
been replaced by 3 x's. If s contains
fewer than 3 characters, returns a copy
of s.

Precondition: s contains only lowercase
letters; it may be empty.

Examples:
'apple' -> 'axxxe'
'banana' -> 'bxxxa'
'preliminary' -> 'pxxxy'
'a' -> 'a'
 """

n=len(s)
if(_________________________):        #1
        return s
else:
        return _____________          #2
```

ii) [1 Point] Assign a value to x so that character ‘A’ is printed out:

```
x=_______
if(x%2==0 and x%5==3):
      print(‘A’)
```

Q7) [2 Points] Write a function lucky_sum(a,b,c) that given 3 int values a, b and c, return their sum. However, if one of the values is 13 then it does not count towards the sum and the values to its right count only if they are even.

Example:
lucky_sum(1, 2, 3) → 6 lucky_sum(1, 2, 13) → 3 lucky_sum(1, 13, 3) → 1
lucky_sum(13, 1, 2) → 2 lucky_sum(13, 4, 2) → 6 lucky_sum(13, 5, 21)→0

Q8) [1 Point] Give an example code of one line that when run would result into a name error.

Q9) [2 Points] Consider the below code. Give the maximum number of frames the code will have at any point of time in the stack space. Show each function call with the argument value passed to it.

```
def f(a):
      if a == 1 or a == 0:
            return 6
      else:
            return f(a-1) + f(a-2)

print(f(3))
```

CSE 101 - Introduction to Programming
Mid Term Lab Exam Set 1
Time: 1 hr

Instructions:

1. Below is the template of the file that you need to create.
2. The name of your python code file must be midterm_s1_2018xxx.py.
3. You will require your IIITD Domain ID and password to upload the file.
4. Use of unfair means will lead to serious consequences as per the college policy.

```
# Midterm Lab Exam Set 1 -  2018
# Name:
# Roll Number:
# Section:
# Group:
# Date:
# You need to implement both the functions given in this module.
#function1
def end_other(s1,s2):
      """ Returns True if either of the strings appear at the very end of
the other string, ignoring upper/lower case differences, i.e., computation
is case-insensitive;
else returns False
      Examples:
      end_other("Hiabc","abc") → True
      end_other("AbC","HiaBc") → True
      end_other("abc","abXabc") → True
      end_other("abc","defx") → False
      """
#function2
def count_code(s3):
      """ Returns the number of times the string code appears anywhere in
the given string, except any letter will be accepted in place of d, i.e.,
'cole' and 'cone' will be counted as well;
if there is no occurrence return 0
      Examples:
      count_code("aaacodebbb") → 1
      count_code("cpdexxcode") → 2
      count_code("cozexxcope") → 2
      count_code("aabbccc") → 0
      """
#print output
print("Ouput1 is " + str(end_other("Hiabc","abc")))
print("Output2 is " + str(count_code("cozexxcope")))
```

CSE 101 - Introduction to Programming
Mid Term Lab Exam Set 2
Time: 1 hr

Instructions:

1. Below is the template of the file that you need to create.
2. The name of your python code file must be midterm_s2_2018xxx.py.
3. You will require your IIITD Domain ID and password to upload the file.
4. Use of unfair means will lead to serious consequences as per the college policy.

```
# Midterm Lab Exam Set 2 -  2018
# Name:
# Roll Number:
# Section:
# Group:
# Date:
# You need to implement both the functions given in this module.
#function1
def end_begin_other(s1,s2):
      """ Returns True if either of the strings appear at the very end and
at the very beginning of the other string, ignoring upper/lower case
differences, i.e., computation is case-insensitive. Else returns False
      Examples:
      end_begin_other("AbCHiabc",abc") → True
      end_begin_other("AbC","ABCHiaBc") → True
      end_begin_other("abc,"aBCabXabc") → True
      end_begin_other("abc","aCCabXabc") → False
      """
#function2
def valid_password(s3):
      """ Returns True when password is valid and False otherwise.
      A password is valid if it satisfies following conditions:
      1. It has minimum 8 characters
      2. The alphabet must be between [a-z]
      3. At least one alphabet should be Upper case, i.e. [A-Z]
      4. At least 1 number or digit between [0-9]
      5. At least 1 character from [_ or @ or $]
      Examples:  valid_password("aaac1@SD") → True
                 valid_password("ASDF12@23") → True
                 valid_password("cope1234") → False
      """
#print output
print("Function1 returns " + str(end_begin_other("abc,aBCabXabc")))
print("Function2 returns " + str(valid_password("ASDF12@23")))
```

CSE 101 - Introduction to Programming
Mid Term Lab Exam Set 3
Time: 1 hr

Instructions:

1. Below is the template of the file that you need to create.
2. The name of your python code file must be midterm_s3_2018xxx.py.
3. You will require your IIITD Domain ID and password to upload the file.
4. Use of unfair means will lead to serious consequences as per the college policy.

```
# Midterm Lab Exam Set 3 -  2018
# Name:
# Roll Number:
# Section:
# Group:
# Date:
# You need to implement both the functions given in this module.

#function1
def count_matchingChars(s1,s2):
    """ Returns the count of matching characters in s1 and s2
considering the single count for the character which have duplicates in
the strings. Computation is case-insensitive. If there is not even a
single char matching returns 0.
Examples:
1. count_matchingChars("bbbbbbba","Abbb") → 2              (a,b matches)
2. count_matchingChars("aabcdddekll12@","bb22llll@k55") → 5
                                                          (b,1,2,@,k match)
3. count_matchingChars("abc","defx") → 0                   No match
    """


#function2
def valid_password(s3):
    """ Returns True when password is valid and False otherwise.
    A password is valid if it satisfies following conditions:
    1. It has minimum 8 characters
    2. The alphabet must be between [a-z]
    3. At least one alphabet should be Upper case, i.e. [A-Z]
    4. At least 1 number or digit between [0-9]
    5. At least 1 character from [_ or @ or $]
    6. It should not be a palindrome string, i.e., reverse of a should
not be equal to a.
```

```
Examples:   valid_password("aaac1@SD") → True
            valid_password("ASDF12@23") → True
            valid_password("cope1234") → False
            valid_password("Aaa12@21aaA") → False
    """

#print output
print("No. Of matching characters are " +
str(count_matchingChars("aabcdddekl112@","bb22llll@k55")))
print("Password check returns value " +str(valid_password("Aaa12@21aaA")))
```

Name: _____ Roll Number:_____
Section: _____ Group: _____

CSE-101, Introduction to Programming
Midterm Re-Exam, 2018
Marks: 25 Time: 60 minutes

Instructions
1. For writing Python code in this exam, we recommend that you draw vertical lines to make your indentation clear. 2. Assume the use of Python3 in all of the questions below.

Q1. Fill in the blanks. The following program checks whether a point (x,y) is strictly inside a square or not. One of the vertices is at the origin and the length of each side is 5 units. [3 marks]

```
x=int(input(" Enter a x coordinate of point"))
y=int(input(" Enter a y coordinate of point"))
# x0,y0 are bottom-left coordinates, i.e., origin and x1,y1 are top-right coordinates of the square.
x1= _____

y1=_____

x0=0

y0=0

if(_____ or _____) :

        print("Outside")
else:
        print("Inside")
```

Q2. Indicate the output if the following script is run: [4 marks]

def F(x,y):
  u = x+2*y
  print (x,y,u)
  return x
x = 1
y = 10
u = 0
print (x,y,u)
y = F(y,x)+F(2*x,y)
print (x,y,u)	

Name: _____ Roll Number:_____

Section: _____ Group: _____

Q3: The function 'not_bad(s)' given a string is assumed to find the first appearance of the substring 'not' and 'bad'. If the 'bad' follows the 'not', it replaces the whole substring starting from 'not' and ending with 'bad', i.e., 'not .....bad', with good. The given function definition is incorrect, correct the line(s) with error. [2 marks]

```
def not_bad(s):
        badindex = s.find('bad')
        notindex = s.find('not')
        if badindex > notindex:
                s= s[ : notindex] + 'good' + s[badindex+3:]
        return s
```

Q4: If following is executed what is the output? [2 marks]

```
x = [10,20,30,40]
y = x
for k in range(4):
        x[k] = y[3-k]
print (x)
```

Q5. What is the output? [4 marks]

```
def foo(a):
        b= True
        for k in range(len(a)):
                b=b and decr(a,k)
        return b

def decr(a,k):
        a[k] = a[k-1]
        return a[k] >= 0
a = [1,2,3,4]
print(foo(a))
print(a)
print(foo(a))
print(a)
```

Name: _____ Roll Number:_____
Section: _____ Group: _____

Q6. Write a python function printPrimes() that takes as parameter an integer n, for n>=2 and prints all the prime numbers starting from 2 to n. [5 marks]

Q7. For the code you have written above show an execution trace for n=6. [5 marks]